United States
Department of
Agriculture

Agricultural
Research
Service

ARS-91

October 1992

GRAMI: A Crop Growth Model That Can Use Remotely Sensed Information

Abstract

Maas, S.J. 1992. GRAMI: A Crop Growth Model That Can Use Remotely Sensed Information. U.S. Department of Agriculture, Agricultural Research Service, ARS 91. 78 pp.

GRAMI is a mathematical model that uses weather and plant canopy observations to simulate growth and yield of gramineous crops. The incorporation of remotely sensed information into mathematical models that simulate crop growth and yield has been a research objective for over a decade. The numerical solution technique in GRAMI allows the user to avoid many of the previous constraints involved in meeting this objective. To produce a simulation, the technique requires the user to input one or more GLAI (green leaf area index) observations, which can be obtained by methods of destructive sampling or remote sensing.

KEYWORDS: computer, crop growth, model, modeling, remote sensing, simulate, simulation

Copies of this publication may be purchased from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

ARS has no additional copies for free distribution.

Contents

Introduction 1
  Model Background 1
  Model Philosophy 3
  Model Requirements 6

Model Listing 8

Model Execution 26

Technical Description of the Model 31
  Crop Growth 31
  Numerical Solution Technique 36
  Input/Output 49

Examples for Various Crops 54
  Spring Wheat 54
  Corn (Maize) 55
  Grain Sorghum 57
  Winter Wheat 57

Summary 61
References 62
Appendix A: Glossary of Variables 65
Appendix B: Listing of CONV_WX 72
Appendix C: Listing of CONV_OB 75
Appendix D: Parameter Default File 78

GRAMI: A Crop Growth Model That Can Use Remotely Sensed Information

Stephan J. Maas

Introduction

This document describes GRAMI, a model designed to demonstrate the use of remotely sensed information in simulating the growth and yield of grain crops. The growth habits of the various gramineous crops (wheat, corn, sorghum, etc.) are sufficiently similar, so a single model can be used for all such crops, with differences among them being accounted for by the values assigned to certain model parameters. Growth is determined using relationships based on crop physiology. The growth functions used in GRAMI are relatively simple, and readers will recognize that more sophisticated approaches can be introduced into the model. GRAMI is a prototype that researchers can use to design operational crop yield estimation and prediction models.

An important objective of this document is to describe a novel numerical technique for using remotely sensed information in the model. Here, "remotely sensed information" refers to a quantification of some crop attribute (such as biomass or leaf area index) obtained through noncontact methods, usually involving the measurement of electromagnetic radiation in specific wavelengths reflected by the crop canopy. The numerical technique in GRAMI was designed to be independent of the source of the remotely sensed data (ground-, aircraft-, or satellite-based instruments). Field data obtained through destructive sampling can also be used in place of remotely sensed data.

Attempts at incorporating remotely sensed information into crop growth simulation models preceding the development of GRAMI are summarized in the following section of this document. The basic ideas that underlie the design of GRAMI are described in the Model Philosophy section. The Model Requirements section outlines the resources needed to run the model. The computer program that constitutes GRAMI is presented in the Model Listing section. A step-by-step description of how to run the program is contained in the Model Execution section. Details of the computational and input/output procedures in GRAMI are presented in the Technical Description of Model section. The Examples for Various Crops section contains results of actual model simulations for growth of spring wheat, corn, grain sorghum, and winter wheat.

Model Background

Few concerted efforts to use remotely sensed information in crop growth models have been made. One of the earliest attempts arose from the development of a grain sorghum growth model at the Blackland Research Center, Temple, TX (Arkin et al. 1976, Vanderlip and Arkin 1977). From this effort, Arkin et al. (1977) recognized that values of model variables could be updated (replaced) with observed values and proposed the concept of a hybrid "spectral-physiological" model capable of using Landsat data. This updating technique was formally described for the model now called SORGF, by Maas and Arkin (1978). The updating technique was used in a study involving 10 fields in Bell

Stephan J. Maas is a plant physiologist, U.S. Department of Agriculture, Agricultural Research Service, Remote Sensing Research Unit, 2413 E. Highway 83, Weslaco, TX 78596-8433.

County, TX (Arkin et al. 1980), but the updating data came from field, and not remotely sensed, observations. Landsat data for these 10 fields were described by Richardson et al. (1982), but the data were not used to update SORGF. The main difficulty in updating SORGF arose from the model's large number of variables, many of which could not be directly evaluated using remotely sensed data.

Another early effort was initiated with the development of a winter wheat model at the Kansas State University Evapotranspiration Laboratory (Hodges and Kanemasu 1977). Since this model used leaf area index (LAI) as a daily input, it was recognized that satellites might be a workable source for these data (Wiegand et al. 1979). Brakke and Kanemasu (1979) used the LAI input technique to estimate winter wheat yields at seven field sites and found correlation coefficients as high as 0.81 between estimated and observed yields. Daily LAI inputs were obtained from curves drawn through multidate Landsat observations. The use of both remotely sensed reflectance and thermal data to determine LAI and soil moisture inputs to KSWHT (a version of the model capable of simulating LAI when inputs were unavailable) was reported by Kanemasu et al. (1982). The study was conducted on irrigation treatment plots at Phoenix, AZ, where remotely sensed observations were made daily with hand-held instruments. The use of both remotely sensed LAI and soil moisture inputs in KSWHT improved the average plot yield estimate from a 34-percent underestimate to a 4-percent overestimate. This model was later used to examine the effects of using different types of LAI information (model-simulated, field-measured, or remotely sensed) on simulated yield estimates (Kanemasu et al. 1985). The studies involving KSWHT verified the hypothesis that crop yield may be accurately estimated through a combination of modeling and remote sensing. However, Kanemasu et al. (1985) concluded that the use of Landsat data as model inputs presented a major operational problem due to the infrequency of observations.

Two more recent efforts (Wanjura and Hatfield 1985, Asrar et al. 1985) utilized simple solar-energy-conversion models and ground-based reflectance data to estimate aboveground dry mass of cotton, soybeans, sunflower, and wheat. Yield was not estimated, and the models were dependent on frequent remotely sensed observations.

Maas et al. (1985) described a corn growth model that used remotely sensed data for both updating and input. Unlike SORGF, the model had only a few variables that required updating. Water stress could be directly evaluated from frequent input of crop water stress index (CWSI) data. Operation of the model was demonstrated using ground-based remotely sensed data from irrigated and dryland field plots.

Recently, Delecolle and Guerif (1988) described a technique for introducing remotely sensed information into AFRCWHEAT, a winter wheat growth and yield simulation model. The methodology was similar to that used earlier in KSWHT, which used daily estimates of LAI as inputs. LAI estimates for AFRCWHEAT came from a statistical model that is driven by temperature and is fit to infrequent observations obtained by remote sensing. When the AFRCWHEAT model was applied to nine durum wheat fields in the Camargue region of France, the prediction error was much lower when remotely sensed LAI estimates were used as compared to when no remotely sensed data were used.

Maas (1988b) described a procedure for using remotely sensed information in conjunction with a crop growth model. The model utilized an iterative numerical solution technique to manipulate an initial condition so that, upon convergence of the numerical solution, the model simulation will fit infrequent values of LAI derived from remotely sensed observations. The model was initially tested using the field and Landsat data from the Bell

County sorghum study (Richardson et al. 1982). A validation study involving 37 grain sorghum fields in Hidalgo County, TX, showed that the model's estimate of overall yield improved from a 31 percent underestimate using no remotely sensed data to a 2 percent overestimate using satellite (Landsat) data. This improvement was statistically significant and resulted from more accurate estimation of individual field yields. The success with this model led to many improvements in the technique, culminating with the development of GRAMI, the model described in this document.

Model Philosophy

Combining Remote Sensing and Modeling. Remote sensing and modeling are techniques that individually have certain strengths and weaknesses for evaluating crop growth and yield. Strengths of remote sensing include the following:

- Remotely sensed observations can provide a quantification of the actual state of the crop during the growing season.
- Using aircraft or satellites, information on crop status can be obtained for many fields in a geographical region by methods that are less labor and material intensive than on-site sampling.

Weaknesses of remote sensing include the following:

- Remotely sensed observations represent discrete time events and indicate little about how the crop achieved its observed state.
- Crop growth and yield are inferred from these discrete observations only through empirical methods and with questionable general application.

Strengths of simulation modeling include the following:

- Models provide a continuous description of the interaction between the crop and its environment over the growing season.
- Models use procedures based on an understanding of crop physiology to calculate growth and yield from environmental conditions. Thus, models should generally be more applicable than empirical techniques are for estimating growth and yield at different locations and time periods.

Weaknesses of simulation modeling include the following:

- Models that rely solely on environmental conditions to simulate growth must contain a considerable amount of detail, since the interactions between plants and their environment can be complex.
- Models generally require detailed on-site observations of environmental data inputs and thus preclude the use of data collected by established regional networks (such as National Weather Service stations).

The basic advantage of combining remote sensing and simulation modeling is that the strengths of one technology make up for the weaknesses of the other. The differences between remote sensing and modeling, however, make the incorporation of remotely sensed information into models more than a trivial exercise.

Incorporating Remotely Sensed Information Into the Model. All crop growth simulation models basically consist of the following: *state variables* (such as leaf area or dry mass), which describe the state of the crop at a given time; *input* or *driving variables* (such as temperature or intercepted solar radiation), which provide the basis for change in the modeled crop; and *parameters*, which determine the response of the state variables to the driving variables. The *initial conditions* (often zero) of the model state variables must be established at the start of the simulation.

The basic components of crop growth models provide avenues for incorporating remotely sensed information into the simulations. Maas (1988a) described four methods of incorporating such information into models: input, updating, reinitialization, and reparameterization. Input is the simplest technique and involves using remotely sensed observations to evaluate model driving variables. Updating involves replacing simulated values of model state variables with values determined from remotely sensed data, in effect providing a new starting point within the growing season each time a simulated value is replaced. Reinitialization and reparameterization involve using an iterative numerical technique to manipulate model initial conditions or parameters so that the resulting simulation fits remotely sensed observations.

Due to the considerable cost of operating aircraft, the timing of satellite overpasses, and the unreliability of cloudless sky conditions, aircraft and satellite observations are typically too infrequent for use in evaluating model inputs. Although values of driving variables for periods between observations can be estimated using interpolation techniques (Brakke and Kanemasu 1979, Kanemasu et al. 1982, Delecolle and Guerif 1988), this practice introduces additional subjectivity and empiricism into the modeling technique. Like all measurements of natural systems, remotely sensed data (particularly those from aircraft or satellites) contain error. Maas (1988a) showed that the effects of measurement error are compounded in model simulations involving input of remotely sensed data. Measurement error also has an undesireable effect when remotely sensed data are used in updating crop growth models. Maas (1988a) showed that the accuracy of the updated model depends almost exclusively on the accuracy of the latest observation and that the effects of errors among observations do not cancel. These characteristics make input and updating inappropriate techniques for use in an operational crop growth and yield model.

Reinitialization and reparameterization are more sophisticated approaches to incorporating remotely sensed information in simulation models. The remotely sensed data are not used directly in computing crop growth, but they are used to direct the manipulation of selected model initial conditions or parameters that affect modeled growth. The resulting growth simulation passes through the observed values in the manner of a best fit, such that the effects of random errors in the observations cancel (Maas 1988a). As few as one observation is effective in significantly improving model performance (Maas 1988b). Recent abstracted (Maas 1987, 1988c; Maas et al. 1988, 1989) and unpublished studies have shown that models combining reinitialization and reparameterization can simulate much of the detail in frequent field observations of crop growth. Based on the favorable experience with these techniques, a combination of reinitialization and reparameterization was implemented in GRAMI.

Accounting for Variations in Crop Growth. Maas (1988a, 1988b) showed that a model containing approximately 10 algorithms can reasonably simulate the qualitative nature (shape) of the crop growth curve over the growing season. The detail in sophisticated crop growth models is required to accurately simulate the quantitative nature (magnitude) of crop growth resulting from various environmental conditions (temperature, solar radiation, etc.) and cultural practices (seeding rate, variety selection, fertilization rate, etc.). If information concerning the magnitude of growth were available during the growing season from observations, much of the detail in crop growth models would be unnecessary. Complex sequences of growth process functions found in sophisticated crop models are replaced in GRAMI by simpler algorithms with parameters evaluated through reparameterization using the observed growth data. Other factors that affect the observed magnitude of growth (cultural practices, diseases, insects, etc.) are also implicitly accounted for in GRAMI by the reparameterization process and by the evaluation of state variable initial conditions through reinitialization based on the growth observations.

Although the GRAMI simulation cannot demonstrate the effects of individual factors on growth, it can approximate actual crop performance using relatively simple inputs. This tradeoff makes GRAMI a poor prototype as a diagnostic model but a good prototype as an operational estimation model.

Like most crop growth simulation models, GRAMI evaluates the variation in crop growth over only the time dimension. Simulated growth is assumed to be a uniform representation of actual growth over some finite spatial scale. Although some nonuniformity in crop growth usually exists within a field, the larger variations that can arise from differing cultural practices (planting date, seeding rate, variety selection, type of tillage, preceding land use, fertilization rate, etc.) dictate that GRAMI cannot directly characterize growth and yield on a spatial scale larger than an individual field. GRAMI operates with a daily time step, which is appropriate for models with this spatial scale (Strand 1981).

The spatial scale of GRAMI restricts the type of remote sensing instruments that can be used for data collection to those that can sample data from individual fields. Fortunately, many different remote sensing systems can provide information for individual fields. Hand-held or vehicle-mounted multiband radiometers can easily provide a number of point measurements within a field, and the measurements can be averaged to produce a value representing field conditions. Such instruments can also be flown in aircraft at an altitude that ensures that the field of view of the sensor is within the dimensions of the field. Scanning radiometers flown in aircraft can produce multispectral images of the earth's surface. Individual field data can be extracted from these images using computer-based analysis systems. Sensors aboard earth-resources satellites (like the Landsat and SPOT series) also can resolve individual fields. Sensors aboard weather satellites (like the NOAA and GOES series) typically cannot resolve individual fields. Therefore, weather satellites cannot directly provide crop information for use in models like GRAMI.

Relating Remotely Sensed Data to Crop Growth. The interpretation of crop status information from remotely sensed data is accomplished by relating plant canopy characteristics to vegetation indices (VIs). VIs are mathematical combinations of surface reflected radiance measured within spectral wave bands selected to discriminate vegetation from the soil background (Kauth and Thomas 1976, Richardson and Wiegand 1977, Jackson et al. 1980, Jackson 1983). VIs have been related to plant canopy characteristics using empirical functions, but there has been little agreement on the general form of these relationships. Recent efforts at evaluating data from many sources (Wiegand and Hatfield 1988) may alleviate this problem.

For use in reinitializing or reparameterizing a model such as GRAMI, remotely sensed data must be related to crop characteristics that appear as state variables in the model. Numerous studies (e.g., Pollock and Kanemasu 1979, Richardson et al. 1982, Daughtry et al. 1983, Hatfield et al. 1985, Wanjura and Hatfield 1985, Redelfs et al. 1987) have shown that the variation in VI values over the growing season closely resembles the variation in green (living) LAI over the same period. During the early part of the growing season, when the canopy of gramineous crops consists primarily of leaf tissue, VIs can also be related to aboveground biomass. Following anthesis, green leaf area decreases through senescence, but the aboveground biomass continues to increase as reproductive organs grow (Richardson et al. 1982). Thus, the early-season correlation between VI and biomass simply reflects the correlation between biomass and green leaf area during this period. As a result, green LAI (GLAI) was selected to be the remotely sensed state variable used in reinitializing and reparameterizing GRAMI.

The incorporation of remotely sensed observations of GLAI into GRAMI has an added benefit with regard to water stress. Physiological studies (Wilson et al. 1980, McCree et al. 1984) indicate that water stress affects plant growth in two ways—through reduced photosynthetic area resulting from decreased leaf expansion or increased leaf senescence and through reduced photosynthetic efficiency per unit leaf area resulting from decreased stomatal conductance. When soil moisture is depleted, leaf expansion is affected before photosynthesis and may be the only effect noted in some cases (Garrity et al. 1984). These water stress effects manifested in the leaf area of the crop are implicitly incorporated into the GRAMI simulation through the remotely sensed observations of GLAI, without having to compute water stress effects from soil moisture or precipitation data.

Applying GRAMI. The manner in which GRAMI simulates crop growth allows it to be directly applied to estimating the yields of individual fields. Average yields over larger geographical areas, such as counties, States, or nations, can be produced by aggregating yield estimates for a number of fields within the area. This technique is currently employed by the U.S. Department of Agriculture to produce State and national crop estimates from on-site field sampling and simple statistical models (Matthews 1985). Because models like GRAMI are weather driven, they could be used during the growing season to predict yield at harvest. If the model is run up to the current day using observed weather data and remotely sensed data, growth for the remaining growing season can be simulated using surrogate daily weather values. Since the actual weather for the remaining growing season is not known with certainty, the model can be run with a number of surrogate weather data sets representing "possible seasons." The resulting distribution of simulated yields can be used to predict the most likely yield at harvest, the probability that the yield will fall within a certain range, and the probability that the yield will be less or greater than a certain value (Arkin et al. 1980).

Model Requirements

The computer program that constitutes GRAMI is written in ANSI Standard FORTRAN 77, a language understood by most scientists involved in computerized crop modeling. The availability of FORTRAN compilers allows GRAMI to be implemented on a wide range of machines, from mainframe computers to personal computers (PCs). Memory requirements for running the model are relatively modest. GRAMI was developed and tested on an AT&T PC 6300 computer system installed with a Microsoft FORTRAN Optimizing Compiler (Version 4.0). The compiled executable file representing the model occupied 61,672 bytes of memory on this system.

GRAMI requires observations of average daily temperature (in degrees Celsius) and accumulated daily photosynthetically active solar irradiance (400-700 nm wavelength band, in MJ m^{-2}) to simulate crop growth. A utility program (CONV_WX) has been provided in appendix B to convert existing weather data sets into input data sets that have the format expected by GRAMI. The user must also supply the planting date of the field to be simulated.

The values of a number of parameters and initial conditions that affect simulated growth or control the numerical solution must be specified to GRAMI. These can be input at the start of a simulation using a disk file like the example provided in appendix D. Values in the disk file may be changed at the discretion of the user by editing the file prior to executing the model or by replacing values in a menu supplied by the model at the start of a simulation. Default values can also be built into the model, but changing these values requires recompiling the program. Methods for determining values of these parameters and initial conditions are described in the Technical Description of Model section.

To make use of the numerical solution technique in GRAMI, the user must supply at least one observed GLAI value for the period during which the crop was grown. The number and the distribution of GLAI observations over the growing season determines which parameters and/or initial conditions will be manipulated in the numerical solution. A utility program (CONV_OB) is provided in appendix C for extracting the required GLAI information from existing data sets and for converting the observations into the format required by the model. Alternately, GLAI values may be input in an interactive manner at the start of the model simulation.

Model Listing

This section contains a listing of the FORTRAN program that constitutes the version of GRAMI developed for use on the personal computer. Minor changes may be required to run this program on larger systems. Such changes should be obvious when the program is compiled. A glossary of variable names appearing in this listing is presented in appendix A.

When first reading this document, the user should briefly scan the model listing to become familiar with the general structure of the program. The user can then proceed to the Model Execution and Technical Description of Model sections for details of the program, referring back to the listing by way of the program line numbers.

```
C     PROGRAM PC-GRAMI Version 1.0 Copy 000
      CHARACTER YES1,YES2,NO1,NO2,ANS,HEDLIN(60),IPLT(71)
      DIMENSION PAR(200),TAVE(200),IDATE(50),OBLAI(50),P(20,3)
      DIMENSION SGDD(200),DGLAI(200),GLAI(200),AGDM(200),TAGDM(200)
      DATA YES1,YES2,NO1,NO2/1HY,1Hy,1HN,1Hn/
C
      WRITE(*,*) '-------------------------------------------------'
      WRITE(*,*) '| PC-Based Version 1.0 of gramineous crop model |'
      WRITE(*,*) '| developed by Stephan J. Maas at the USDA/ARS  |'
      WRITE(*,*) '| Subtropical Agric. Res. Lab., Weslaco, Texas  |'
      WRITE(*,*) '| Reference: Maas (1992) USDA ARS-91.           |'
      WRITE(*,*) '-------------------------------------------------'
      WRITE(*,*) '  '
      WRITE(*,*) 'SIMULATION SUMMARY WILL BE WRITTEN TO UNIT 20'
      OPEN(20,STATUS='NEW',FILE='   ')
      WRITE(*,*) 'DO YOU WANT TO WRITE A HEADLINE ON THIS SUMMARY ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
 5000 FORMAT(A1)
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) THEN
         WRITE(*,*) 'ENTER CONTENTS (MAX OF 60 CHARACTERS) OF THE',
     2              ' HEADLINE-'
         READ(*,5010)   (HEDLIN(I), I=1,60)
         WRITE(20,5010) (HEDLIN(I), I=1,60)
 5010    FORMAT(60A1)
      ENDIF
C
      WRITE(*,*) 'WEATHER DATA (MAX OF 200 DAYS) WILL BE READ',
     2           ' FROM UNIT 10'
      OPEN(10,STATUS='OLD',FILE='   ')
      NDAYS = 0
         DO 5020 I=1, 200
         READ(10,*,END=5030) IDAY,TAVE(I),PAR(I)
         IF (I .EQ. 1) ISTRT = IDAY
 5020    NDAYS = NDAYS + 1
C
 5030 WRITE(*,*) 'ENTER THE YEAR AND DAY OF YEAR OF PLANTING'
      WRITE(*,*) '(SEPARATED BY A COMMA)-'
      READ(*,*) IYEAR,ISOW
      WRITE(20,5040) ISTRT,ISOW
 5040 FORMAT('WEATHER DATA STARTS ON DAY',I5,', PLANTING ON DAY',I5)
      NYEAR = 365
      IF ((IYEAR/4)*4 .EQ. IYEAR) NYEAR = 366
      IF (ISOW .LT. ISTRT) ISOW = ISOW + NYEAR
C
      WRITE(*,*) 'GROWTH OF THE FOLLOWING CROPS CAN BE SIMULATED-'
      WRITE(*,*) '  1   Spring Wheat'
      WRITE(*,*) '  2   Corn (Maize)'
      WRITE(*,*) '  3   Grain Sorghum'
      WRITE(*,*) '  4   A Different Crop'
      WRITE(*,*) 'ENTER THE NO. OF THE CROP YOU WANT TO SIMULATE-'
      WRITE(*,*) '(NOTE- IF YOU ENTER NO. 4, YOU MUST INPUT THE'
```

```
      WRITE(*,*) ' PARAMETER DEFAULT VALUES FROM A DISK FILE)'
      READ(*,*) JCROP
C
C     Built-in default parameter values (P) for the various crops-
      DATA P/150.,1450.,2100.,0.,.010,.60,2.5,.024,.90,.040,.003,
     2       70.00,.001,.0001,4.00,.0001,.00001,0.5,.01,.05,
     3       70.,1000.,1800.,10.,.010,.35,3.5,.024,.50,.050,.003,
     4       50.00,.001,.0001,4.00,.0001,.00001,0.5,.01,.05,
     5       60.,900.0,1600.,10.,.010,.30,3.5,.024,.75,.067,.003,
     6       50.00,.001,.0001,4.00,.0001,.00001,0.5,.01,.05/
C
      IF (JCROP .NE. 4) THEN
         WRITE(*,*) 'DO YOU WANT TO ENTER PARAMETER DEFAULT VALUES',
     2              ' FROM A DISK FILE ?'
         WRITE(*,*) '(ENTER Y OR N)-'
         READ(*,5000) ANS
      ELSE
         ANS = YES1
      ENDIF
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) THEN
         JCROP = 1
         WRITE(*,*) 'DISK FILE WILL BE READ FROM UNIT 30-'
         OPEN(30,STATUS='OLD',FILE='   ')
         READ(30,*)
         READ(30,*)
         DO 5050 I=1, 20
 5050    READ(30,*) P(I,JCROP)
      ENDIF
C
      WRITE(*,*) 'MODEL PARAMETER DEFAULT VALUES FOR THIS CROP ARE-'
 5060 WRITE(*,5070) (P(I,JCROP), I=1,10)
 5070 FORMAT('    1  Sum Deg-Days to Emergence (SDDEM)=',F10.1/
     2       '    2  Sum Deg-Days to Anthesis  (SDDAN)=',F10.1/
     3       '    3  Sum Deg-Days to Maturity  (SDDPM)=',F10.1/
     4       '    4  Base Temp for Deg-Days    (BASET)=',F10.1/
     5       '    5  Initial Leaf Area Index   (XGLAI)=',F10.5/
     6       '    6  Light Extinction Coefficient (EX)=',F10.3/
     7       '    7  DM/PAR Conversion Ratio      (EF)=',F10.3/
     8       '    8  Specific Leaf Area          (SLA)=',F10.5/
     9       '    9  Grain DM Partitioning Ratio  (YF)=',F10.3/
     1       '   10  Initial Value for Parameter A (A)=',F10.5)
      WRITE(*,5080) (P(I,JCROP), I=11,20)
 5080 FORMAT('   11  Initial Value for Parameter B (B)=',F10.5/
     2       '   12  Init Value for Leaf Lifespan  (C)=',F10.1/
     3       '   13  Init Iter Interval for A  (XDELA)=',F10.5/
     4       '   14  Init Iter Interval for B  (XDELB)=',F10.6/
     5       '   15  Init Iter Interval for C  (XDELC)=',F10.1/
     6       '   16  Converg Criterion for A   (CONVA)=',F10.6/
     7       '   17  Converg Criterion for B   (CONVB)=',F10.6/
     8       '   18  Converg Criterion for C   (CONVC)=',F10.2/
     9       '   19  Conv Criterion for XGLAI  (CONVG)=',F10.4/
     1       '   20  Average Tolerable Error   (AVTOL)=',F10.4)
      WRITE(*,*) 'DO YOU WANT TO CHANGE A DEFAULT VALUE ?'
      WRITE(*,*) '(ENTER Y OR N)-'
```

```
      READ(*,5000) ANS
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) THEN
         WRITE(*,*) 'ENTER THE DEFAULT NO. (1-20) AND NEW VALUE',
     2              ' (SEPARATED BY A COMMA)-'
         READ(*,*) IDEF,XDEF
         P(IDEF,JCROP) = XDEF
         WRITE(*,5090) IDEF
 5090    FORMAT(' DEFAULT VALUE NO.',I3,' CHANGED-')
         GO TO 5060
      ENDIF
C
C     Load default values (P) for model parameters-
      SDDEM = P( 1,JCROP)
      SDDAN = P( 2,JCROP)
      SDDPM = P( 3,JCROP)
      BASET = P( 4,JCROP)
      XGLAI = P( 5,JCROP)
      EX    = P( 6,JCROP)
      EF    = P( 7,JCROP)
      SLA   = P( 8,JCROP)
      YF    = P( 9,JCROP)
      A     = P(10,JCROP)
      B     = P(11,JCROP)
      C     = P(12,JCROP)
      XDELA = P(13,JCROP)
      XDELB = P(14,JCROP)
      XDELC = P(15,JCROP)
      CONVA = P(16,JCROP)
      CONVB = P(17,JCROP)
      CONVC = P(18,JCROP)
      CONVG = P(19,JCROP)
      AVTOL = P(20,JCROP)
      WRITE(20,5100) SDDEM,SDDAN,SDDPM,BASET
 5100 FORMAT('STARTING VALUES FOR SDDEM, SDDAN, SDDPM AND',
     2       ' BASET:'/4F8.1)
      WRITE(20,5110) EX,EF,SLA,YF,A,B,C
 5110 FORMAT('STARTING VALUES FOR EX, EF, SLA, YF, A, B AND',
     2       ' C:'/4F7.4,F9.6,F9.6,F7.1)
      WRITE(20,5120) XGLAI
 5120 FORMAT('INITIAL GLAI VALUE:',F12.8)
C
      MODEL = 0
      WRITE(*,*) 'DO YOU WANT TO INPUT OBSERVED GLAI VALUES ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
      IF (ANS.EQ.NO1 .OR. ANS.EQ.NO2) GO TO 5480
      WRITE(*,*) 'HOW DO YOU WANT TO INPUT THEM ?'
      WRITE(*,*) '(ENTER 1 FOR DISK FILE, 2 FOR KEYBOARD)-'
      READ(*,*) KENTER
      IF (KENTER .EQ. 1) THEN
         WRITE(*,*) 'OBSERVATIONS WILL BE READ FROM UNIT 40'
         OPEN(40,STATUS='OLD',FILE='   ')
         NOBS = 0
            DO 5130 I=1, 50
```

```
            READ(40,*,END=5160) IDATE(I),OBLAI(I)
 5130       NOBS = NOBS + 1
      ELSE
         WRITE(*,*) 'ENTER NUMBER OF GLAI OBSERVATIONS (MAX OF 50)-'
         READ(*,*) NOBS
            DO 5150 I=1, NOBS
            WRITE(*,5140) I
 5140       FORMAT(' ENTER DAY OF YEAR AND GLAI VALUE (SEPARATED',
     2             ' BY A COMMA)'/' FOR OBSERVATION NO.',I3,'-')
 5150       READ(*,*) IDATE(I),OBLAI(I)
      ENDIF
 5160 EMAX = 0.
      WRITE(20,*) '      DAY     OBS LAI'
         DO 5180 I=1, NOBS
         WRITE(20,5170) IDATE(I),OBLAI(I)
 5170    FORMAT(I10,F12.3)
         IF (IDATE(I) .LT. ISTRT) IDATE(I) = IDATE(I) + NYEAR
 5180    EMAX = EMAX + OBLAI(I)
C
      WRITE(*,*) 'SHOULD NUMERICAL SOLUTION TECHNIQUE BE USED ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
      IF (ANS.EQ.NO1 .OR. ANS.EQ.NO2) GO TO 5480
      WRITE(20,5190) XDELA,XDELB,XDELC,CONVA,CONVB,CONVC,CONVG,AVTOL
 5190 FORMAT('VALUES FOR XDELA, XDELB, XDELC, CONVA, CONVB,'
     2       ' CONVC, CONVG AND AVTOL:'/2F9.6,F7.1,2F9.6,F7.2,2F9.6)
C
C     Set limits for numerical solution parameters-
      AMIN = 0.
      AMAX = 1.
      BMIN = 0.
      BMAX = 0.1
      CMIN = 0.
      CMAX = FLOAT(NDAYS)
      TOLER  = NOBS*AVTOL
      MAXTRY = 25
C
C     Suggest which form of numerical model should be used-
      IF (NOBS .EQ. 0) GO TO 5230
      MODEL = 1
C     Determine approx dates of max LAI (JMX) and maturity (JPM)-
      SDD = 0.
         DO 5200 I=1, NDAYS
         IDAY = I + ISTRT -1
         IF (IDAY .LT. ISOW) GO TO 5200
         SDD = SDD + AMAX1(TAVE(I)-BASET,0.)
         IF (SDD .GT. SDDPM) GO TO 5210
         JPM = IDAY
         IF (SDD .LT. SDDEM) GO TO 5200
         SPF = A*EXP( B*(SDD-SDDEM) )
         IF (SPF .GT. 1.) GO TO 5200
         JMX = IDAY
 5200    CONTINUE
C     Count number of LAI observations (NOBSX) between JMX and JPM-
```

```
 5210 NOBSX = 0
      MXDAY = JMX
      MNDAY = JPM
         DO 5220 J=1, NOBS
         IF (IDATE(J) .LE. JMX) GO TO 5220
         NOBSX = NOBSX + 1
         IF (IDATE(J) .GT. MXDAY) MXDAY = IDATE(J)
         IF (IDATE(J) .LT. MNDAY) MNDAY = IDATE(J)
 5220    CONTINUE
      IF (NOBSX .LT. 2) GO TO 5230
      DIST = FLOAT(MXDAY-MNDAY) / (JPM-JMX)
      IF (DIST .LT. 0.333) GO TO 5230
      MODEL = 2
C     Determine if higher-order model might be used-
      IF (NOBS-NOBSX .EQ. 0) GO TO 5230
      MODEL = 3
 5230 WRITE(*,*) 'THERE ARE 5 FORMS OF THE NUMERICAL MODEL-'
      WRITE(*,*) '  Form 0-   No numerical solution used'
      WRITE(*,*) '  Form 1-   Simple solution (only XGLAI',
     2           ' manipulated)'
      WRITE(*,*) '  Form 2-   XGLAI and parameter C manipulated'
      WRITE(*,*) '  Form 3-   XGLAI and parameters B and C',
     2           ' manipulated'
      WRITE(*,*) '  Form 4-   Full solution (XGLAI, A, B and C',
     2           ' manipulated)'
      WRITE(*,5240) MODEL
 5240 FORMAT(' PRELIMINARY EXAMINATION OF THE GLAI OBSERVATIONS',
     2       ' SUGGESTS'/' THAT MODEL FORM',I3,' SHOULD BE USED')
      WRITE(*,*) 'DO YOU WANT TO USE A DIFFERENT FORM THAN THIS ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) THEN
         WRITE(*,*) 'ENTER THE DESIRED MODEL FORM (0,1,2,3 OR 4)-'
         READ(*,*) MODEL
         WRITE(20,5250) MODEL
 5250    FORMAT('MODEL FORM',I3,' SELECTED FOR NUMERICAL SOLUTION')
      ENDIF
      IF (MODEL .EQ. 0) GO TO 5480
C
      PAUSE 'PRESS Enter KEY TO START ITERATIVE NUMERICAL SOLUTION'
      NITER = 0
      GO TO (5470,5400,5330,5260), MODEL
C     Set starting values for parameter A solution-
 5260 DELA = XDELA
      ITA = 0
      KAPAB = 0
      GO TO 5290
C
C     Start Parameter A Reparameterization   ...................
C
 5270 ITA = ITA + 1
      ASTB = B
      ASTC = C
      ASTG = XGLAI
```

```
      IF (ITA .GT. MAXTRY) THEN
         WRITE(*,5280) MAXTRY
 5280    FORMAT(' NUMBER OF A ITERATIONS EXCEEDS LIMIT (',I3,')-'/
     2          ' SOLN STOPS WITH A VALUE FROM LAST ITERATION')
         GO TO 5760
      ENDIF
      IF (KAPAB .EQ. 1) GO TO 5310
C     Bracketing Search-
      IF (DELA .GT. 0.) THEN
         A = A + DELA*ITA
      ELSE
         A = AMAX1(A+DELA*ITA,A/2.)
      ENDIF
      IF (A.GT.AMAX .OR. A.LT.AMIN) THEN
         WRITE(*,*) 'VALUE OF A IS OUTSIDE PRESCRIBED LIMITS-'
         WRITE(*,*) 'SOLN STOPS WITH A VALUE FROM LAST ITERATION'
         GO TO 5760
      ENDIF
 5290 WRITE(*,5300) ITA,A
 5300 FORMAT(' A ITERATION',I4,', A =',F9.6,' (BRACKETING SEARCH)')
      GO TO 5330
C     Parabolic Interpolation-
 5310 CALL PARAB(AP1,AP2,AP3,AP4,AE1,AE2,AE3)
      A = AP4
      WRITE(*,5320) ITA,A
 5320 FORMAT(' A ITERATION',I4,', A =',F9.6,
     2       ' (PARABOLIC INTERPOLATION)')
C     Set starting values for parameter B solution-
 5330 DELB = XDELB
      ITB = 0
      KBPAB = 0
      GO TO 5360
C
C     Start Parameter B Reparameterization   ....................
C
 5340 ITB = ITB + 1
      BSTC = C
      BSTG = XGLAI
      IF (ITB .GT. MAXTRY) THEN
         WRITE(*,5350) MAXTRY
 5350    FORMAT(3X,' NUMBER OF B ITERATIONS EXCEEDS LIMIT (',I3,')-'
     2          /3X,' SOLN STOPS WITH B VALUE FROM LAST ITERATION')
         GO TO 5700
      ENDIF
      IF (KBPAB .EQ. 1) GO TO 5380
C     Bracketing Search-
      IF (DELB .GT. 0.) THEN
         B = B +DELB*ITB
      ELSE
         B = AMAX1(B+DELB*ITB,B/2.)
      ENDIF
      IF (B.GT.BMAX .OR. B.LT.BMIN) THEN
         WRITE(*,*) '   VALUE OF B IS OUTSIDE PRESCRIBED LIMITS-'
         WRITE(*,*) '   SOLN STOPS WITH B VALUE FROM LAST ITERATION'
```

```
         GO TO 5700
      ENDIF
 5360 WRITE(*,5370) ITB,B
 5370 FORMAT(3X,' B ITERATION',I4,', B =',F9.6,
     2        ' (BRACKETING SEARCH)')
      GO TO 5400
C     Parabolic Interpolation-
 5380 CALL PARAB(BP1,BP2,BP3,BP4,BE1,BE2,BE3)
      B = BP4
      WRITE(*,5390) ITB,B
 5390 FORMAT(3X,' B ITERATION',I4,', B =',F9.6,
     2        ' (PARABOLIC INTERPOLATION)')
C     Set starting values for parameter C solution-
 5400 DELC = XDELC
      ITC = 0
      KCPAB = 0
      GO TO 5430
C
C     Start Parameter C Reparameterization   ....................
C
 5410 ITC = ITC + 1
      CSTG = XGLAI
      IF (ITC .GT. MAXTRY) THEN
         WRITE(*,5420) MAXTRY
 5420    FORMAT(6X,' NUMBER OF C ITERATIONS EXCEEDS LIMIT (',I3,
     2   ')-'/6X,' SOLN STOPS WITH C VALUE FROM LAST ITERATION')
         GO TO 5640
      ENDIF
      IF (KCPAB .EQ. 1) GO TO 5450
C     Bracketing Search-
      IF (DELC .GT. 0.) THEN
         C = C +DELC*ITC
      ELSE
         C = AMAX1(C+DELC*ITC,C/2.)
      ENDIF
      IF (C.GT.CMAX .OR. C.LT.CMIN) THEN
         WRITE(*,*) '      VALUE OF C IS OUTSIDE PRESCRIBED LIMITS-'
         WRITE(*,*) '      SOLN STOPS WITH C VALUE FROM LAST',
     2              ' ITERATION'
         GO TO 5640
      ENDIF
 5430 WRITE(*,5440) ITC,C
 5440 FORMAT(6X,' C ITERATION',I4,', C =',F9.2,
     2        ' (BRACKETING SEARCH)')
      GO TO 5470
C     Parabolic Interpolation-
 5450 CALL PARAB(CP1,CP2,CP3,CP4,CE1,CE2,CE3)
      C = CP4
      WRITE(*,5460) ITC,C
 5460 FORMAT(6X,' C ITERATION',I4,', C =',F9.2,
     2        ' (PARABOLIC INTERPOLATION)')
C     Set starting values for XGLAI solution-
 5470 ITG = 0
 5480 EPOS = 0.
```

```
      ENEG = 0.
      NITER = NITER + 1
      XAGDM = XGLAI/SLA
      YIELD = 0.
C
C     Start Daily Loop  ----------------------------------------
C
      DO 5560 I=1, NDAYS
      IDAY = I + ISTRT -1
C     Set default values for state variables for this day-
      SGDD(I) = 0.
      GLAI(I) = XGLAI
      TAGDM(I) = XAGDM
      AGDM(I)  = XAGDM
C     Determine if crop is between emergence and maturity-
      IF (IDAY .LE. ISOW) GO TO 5500
      SGDD(I) = SGDD(I-1) + AMAX1(TAVE(I)-BASET,0.)
      IF (SGDD(I) .LE. SDDEM) GO TO 5500
      IF (SGDD(I) .GT. SDDPM) THEN
         IF (MODEL .EQ. 0) THEN
            IEND = I -1
            GO TO 5790
         ELSEIF (IDAY .GT. IDATE(NOBS)) THEN
            IEND = I -1
            GO TO 5570
         ENDIF
      ENDIF
C     Calculate absorbed PAR, APAR-
      APAR = PAR(I)*(1. -EXP(-EX*GLAI(I-1)))
C     Calculate daily dry mass production, DAGDM-
      DAGDM = EF*APAR
      TAGDM(I) = TAGDM(I-1) + DAGDM
C     Calculate daily leaf area production, DGLAI-
      SPF = A* EXP(B*( SGDD(I)-SDDEM ))
      IF (SPF .LT. 1.) THEN
         DGLAI(I) = SLA*( 1.-SPF )*DAGDM
      ELSE
         DGLAI(I) = 0.
      ENDIF
C     Calculate GLAI and AGDM including senescence-
      AGDM(I) = TAGDM(I)
      GLAI(I) = 0.
         DO 5490 II=1, I
         IF (SGDD(II) .LT. SDDEM) GO TO 5490
         IF (FLOAT(I-II) .LT. C) THEN
            GLAI(I) = GLAI(I) + DGLAI(II)
         ELSE
            AGDM(I) = AGDM(I) - DGLAI(II)/SLA
         ENDIF
 5490    CONTINUE
C     Calculate daily contribution to yield-
      IF (SGDD(I) .LE. SDDAN) GO TO 5500
      IF (SGDD(I) .LE. SDDPM) YIELD = YIELD + YF*DAGDM
C
```

```
C     Start Error Function Section   ============================
C
 5500 IF (I .EQ. NDAYS) IEND = I
      IF (MODEL .EQ. 0) GO TO 5560
         DO 5510 II=1, NOBS
C        Check if there is a GLAI observation on this day-
         IF (IDAY .NE. IDATE(II)) GO TO 5510
C        Calculate error between observed and simulated GLAI-
         ERROR = OBLAI(II) - GLAI(I)
         IF (ERROR .GT. 0.) THEN
            EPOS = EPOS + ERROR
         ELSE
            ENEG = ENEG - ERROR
         ENDIF
 5510    CONTINUE
      IF (IDAY .NE. IDATE(NOBS)) GO TO 5560
C     Evaluate goodness-of-fit statistic, E-
      E = EPOS - ENEG
C
C     End Error Function Section   ===============================
C
C     Start XGLAI Reinitialization Section   ....................
C
      WRITE(*,5520) ITG,XGLAI
 5520 FORMAT(9X,' XGLAI ITERATION',I4,', XGLAI =',F11.9)
      IF (ABS(E) .LE. CONVG) THEN
         WRITE(*,5530) EPOS,ENEG
 5530    FORMAT(9X,' SOLUTION CONVERGED (EPOS =',F7.2,
     2             ', ENEG =',F7.2,')'/)
         GO TO 5560
      ELSE
         WRITE(*,5540) EPOS,ENEG
 5540    FORMAT(9X,' SOLUTION NOT CONVERGED (EPOS =',F7.2,
     2             ', ENEG =',F7.2,')')
      ENDIF
      ITG = ITG + 1
      IF (ITG .GT. MAXTRY) THEN
         WRITE(*,5550) MAXTRY
 5550    FORMAT(9X,' NUMBER OF XGLAI ITERATIONS EXCEEDS LIMIT (',
     2   I3,')-'/9X,' SOLN STOPS WITH XGLAI VALUE FROM LAST ITER')
         GO TO 5560
      ENDIF
      IF (ITG .EQ. 1) THEN
C        Rough guess of new XGLAI value on first iteration-
         OLDE = E
         OLDG = XGLAI
         IF (E .GT. 0.) THEN
            XGLAI = XGLAI*(1. +E/EMAX)
         ELSE
            XGLAI = XGLAI/(1. -E/EMAX)
         ENDIF
      ELSE
C        Secant Method on succeeding iterations-
         SLOPE = (XGLAI-OLDG) / (E-OLDE)
```

```
            YINTC = XGLAI -E*SLOPE
            OLDE = E
            OLDG = XGLAI
            XGLAI = YINTC
            IF (XGLAI .LE. 0.) XGLAI = OLDG/2.
         ENDIF
         GO TO 5480
C
C        End XGLAI Reinitialization Section   ......................
C
 5560    CONTINUE
C
C        End Daily Loop   ------------------------------------------
C
 5570    IF (MODEL .EQ. 0) GO TO 5790
         EC = EPOS + ENEG
         IF (MODEL .EQ. 1) GO TO 5760
         IF (EC .LE. TOLER) THEN
C           Total error is less than tolerable amount-
            WRITE(*,5580) EC,TOLER
 5580       FORMAT(' TOTAL ERROR (',F7.3,') IS LESS THAN TOLERANCE (',
     2      F7.3,')-'/' DO YOU WANT TO ACCEPT THIS AS THE SOLUTION ?'
     3      /' (ENTER Y OR N)-')
            READ(*,5000) ANS
            IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) GO TO 5760
         ENDIF
         IF (KCPAB .EQ. 1) GO TO 5610
C        Bracketing Search-
         IF (ITC .EQ. 0) THEN
            CP1 = C
            CE1 = EC
            GO TO 5410
         ENDIF
         IF (ITC .EQ. 1) THEN
C           Determine search direction-
            IF (EC .LT. CE1) THEN
               CP2 = C
               CE2 = EC
            ELSE
               CALL REVERS(CP1,CP2,CE1,CE2,C,EC)
               C = C -DELC
               XGLAI = CSTG
               DELC = -DELC
            ENDIF
            GO TO 5410
         ENDIF
C        Determine if solution is bracketed-
         IF (ITC .EQ. 2) THEN
            CP3 = C
            CE3 = EC
         ELSE
            CALL REPLAC(CP1,CP2,CP3,CE1,CE2,CE3,C,EC)
         ENDIF
         IF (CE3 .GT. CE2) THEN
```

```
            WRITE(*,5590) C,EC
 5590       FORMAT(6X,' C SOLUTION BRACKETED BY C =',F9.2,
     2                '  (EC =',F7.3,')'/)
            KCPAB = 1
            XGLAI = CSTG
         ELSE
            WRITE(*,5600) C,EC
 5600       FORMAT(6X,' C SOLUTION NOT BRACKETED BY C =',F9.2,
     2                '  (EC =',F7.3,')')
         ENDIF
         GO TO 5410
C        Parabolic Interpolation-
 5610    IF (ABS(CP4-CP2) .GT. CONVC) THEN
            WRITE(*,5620) C,EC
 5620       FORMAT(6X,' C SOLUTION NOT CONVERGED FOR C =',F9.2,
     2                '  (EC =',F7.3,')')
            CE4 = EC
            CALL RESEL3(CP1,CP2,CP3,CP4,CE1,CE2,CE3,CE4)
            GO TO 5410
         ELSE
            WRITE(*,5630) C,EC
 5630       FORMAT(6X,' C SOLUTION CONVERGED FOR C =',F9.2,
     2                '  (EC =',F7.3,')'/)
         ENDIF
C
C        End Parameter C Reparameterization   ......................
C
 5640    EB = EC
         IF (MODEL .EQ. 2) GO TO 5760
         IF (KBPAB .EQ. 1) GO TO 5670
C        Bracketing Search-
         IF (ITB .EQ. 0) THEN
            BP1 = B
            BE1 = EB
            GO TO 5340
         ENDIF
         IF (ITB .EQ. 1) THEN
C           Determine search direction-
            IF (EB .LT. BE1) THEN
               BP2 = B
               BE2 = EB
            ELSE
               CALL REVERS(BP1,BP2,BE1,BE2,B,EB)
               B = B -DELB
               C = BSTC
               XGLAI = BSTG
               DELB = -DELB
            ENDIF
            GO TO 5340
         ENDIF
C        Determine if solution is bracketed-
         IF (ITB .EQ. 2) THEN
            BP3 = B
            BE3 = EB
```

```
      ELSE
         CALL REPLAC(BP1,BP2,BP3,BE1,BE2,BE3,B,EB)
      ENDIF
      IF (BE3 .GT. BE2) THEN
         WRITE(*,5650) B,EB
 5650    FORMAT(3X,' B SOLUTION BRACKETED BY B =',F9.6,
     2             '  (EB =',F7.3,')'/)
         KBPAB = 1
         C = BSTC
         XGLAI = BSTG
      ELSE
         WRITE(*,5660) B,EB
 5660    FORMAT(3X,' B SOLUTION NOT BRACKETED BY B =',F9.6,
     2             '  (EB =',F7.3,')')
      ENDIF
      GO TO 5340
C     Parabolic Interpolation-
 5670 IF (ABS(BP4-BP2) .GT. CONVB) THEN
         WRITE(*,5680) B,EB
 5680    FORMAT(3X,' B SOLUTION NOT CONVERGED FOR B =',F9.6,
     2             '  (EB =',F7.3,')')
         BE4 = EB
         CALL RESEL3(BP1,BP2,BP3,BP4,BE1,BE2,BE3,BE4)
         GO TO 5340
      ELSE
         WRITE(*,5690) B,EB
 5690    FORMAT(3X,' B SOLUTION CONVERGED FOR B =',F9.6,
     2             '  (EB =',F7.3,')'/)
      ENDIF
C
C     End Parameter B Reparameterization   ......................
C
 5700 EA = EB
      IF (MODEL .EQ. 3) GO TO 5760
      IF (KAPAB .EQ. 1) GO TO 5730
C     Bracketing Search-
      IF (ITA .EQ. 0) THEN
         AP1 = A
         AE1 = EA
         GO TO 5270
      ENDIF
      IF (ITA .EQ. 1) THEN
C        Determine search direction-
         IF (EA .LT. AE1) THEN
            AP2 = A
            AE2 = EA
         ELSE
            CALL REVERS(AP1,AP2,AE1,AE2,A,EA)
            A = A -DELA
            B = ASTB
            C = ASTC
            XGLAI = ASTG
            DELA = -DELA
         ENDIF
```

```
         GO TO 5270
      ENDIF
C     Determine if solution is bracketed-
      IF (ITA .EQ. 2) THEN
         AP3 = A
         AE3 = EA
      ELSE
         CALL REPLAC(AP1,AP2,AP3,AE1,AE2,AE3,A,EA)
      ENDIF
      IF (AE3 .GT. AE2) THEN
         WRITE(*,5710) A,EA
 5710    FORMAT(' A SOLUTION BRACKETED BY A =',F9.6,
     2             '  (EA =',F7.3,')'/)
         KAPAB = 1
         B = ASTB
         C = ASTC
         XGLAI = ASTG
      ELSE
         WRITE(*,5720) A,EA
 5720    FORMAT(' A SOLUTION NOT BRACKETED BY A =',F9.6,
     2             '  (EA =',F7.3,')')
      ENDIF
      GO TO 5270
C     Parabolic Interpolation-
 5730 IF (ABS(AP4-AP2) .GT. CONVA) THEN
         WRITE(*,5740) A,EA
 5740    FORMAT(' A SOLUTION NOT CONVERGED FOR A =',F9.6,
     2             '  (EA =',F7.3,')')
         AE4 = EA
         CALL RESEL3(AP1,AP2,AP3,AP4,AE1,AE2,AE3,AE4)
         GO TO 5270
      ELSE
         WRITE(*,5750) A,EA
 5750    FORMAT(' A SOLUTION CONVERGED FOR A =',F9.6,
     2             '  (EA =',F7.3,')'/)
      ENDIF
C
C     End Parameter A Reparameterization   ......................
C
 5760 WRITE(*,*) 'ITERATIVE NUMERICAL SOLUTION IS FINISHED'
      EPERC = 100.*(EMAX-EC)/EMAX
      WRITE(20,5770) A,B,C,XGLAI
 5770 FORMAT(/'VALUES FOR A, B, C AND XGLAI AT CONVERGENCE:'/
     2       2F9.6,F7.1,F11.9)
      WRITE(20,5780) EC,NITER,EPERC
 5780 FORMAT('   TOTAL ERROR =',F7.3,', NUMBER OF ITERATIONS',
     2       ' REQUIRED FOR CONV =',I5/'   PERCENT OF TOTAL',
     3       ' OBSERVED GLAI EXPLAINED =',F6.1)
 5790 YIELD = YIELD*10.
      WRITE(20,5800) YIELD
 5800 FORMAT(/'ESTIMATED GRAIN YIELD (KG/HA) =',F7.0)
C
      WRITE(*,*) 'DO YOU WANT TO PRODUCE A ROUGH GRAPH OF THE MODEL'
      WRITE(*,*) 'SIMULATION WRITTEN TO THE OUTPUT FILE ?'
```

```
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
      IF (ANS.EQ.NO1 .OR. ANS.EQ.NO2) GO TO 5930
      DATA ILINE/1H-/,ICAT/1H¦/,IBLNK/1H /,IGLAI/1H+/,IAGDM/1H*/,
     2     IOBS/1HO/,IANTH/1HA/,IPMAT/1HM/
      WRITE(20,5810) IGLAI,IAGDM,IAGDM,IOBS,IANTH,IPMAT
 5810 FORMAT(///'ROUGH PLOT OF GROWTH SIMULATION RESULTS-'/
     2       5X,'SYMBOLS:'/10X,A1,'  SIMULATED GLAI'/
     3       10X,A1,'  SIMULATED DRY MASS WITH SENESCENCE'/
     4       10X,A1,'  SIMULATED DRY MASS WITHOUT SENESCENCE'/
     5       10X,A1,'  OBSERVED GLAI'/
     6       10X,A1,'  SIMULATED ANTHESIS DAY'/
     7       10X,A1,'  SIMULATED MATURITY DAY'/
     8       /5X,'X-AXIS UNITS ARE DAYS AFTER PLANTING (DAP)'/)
      WRITE(20,5820)
 5820 FORMAT(33X,'LEAF AREA INDEX'/1X,'DAP 0',9X,'2',9X,'4',
     2       9X,'6',9X,'8',9X,'10',8X,'12',8X,'14')
      KANTH = 0
      KPMAT = 0
      ILAST = IEND + 5
C     Start loop to plot results for each day-
      DO 5910 I=1, ILAST
      IDAY = I + ISTRT -1
      IF (IDAY .LT. ISOW) GO TO 5910
      JDAP = IDAY -ISOW
         DO 5830 II=1, 71
 5830    IPLT(II) = IBLNK
      IF (JDAP.EQ.0 .OR. I.EQ.ILAST) THEN
         DO 5840 II=1, 71
 5840    IPLT(II) = ILINE
         DO 5850 J=1, 8
         II = 10*(J-1) +1
 5850    IPLT(II) = ICAT
         GO TO 5880
      ENDIF
      IPLT( 1) = ICAT
      IPLT(71) = ICAT
      IF ((JDAP/10)*10 .EQ. JDAP) THEN
         IPLT( 1) = ILINE
         IPLT(71) = ILINE
      ENDIF
      IF (I .GT. IEND) GO TO 5880
      IF (SGDD(I) .LT. SDDEM) GO TO 5860
      II = IFIX(TAGDM(I)*10./1000. +.5) +1
      IPLT(II) = IAGDM
      II = IFIX(AGDM(I)*10./1000. +.5) +1
      IPLT(II) = IAGDM
      II = IFIX(GLAI(I)*5. +.5) +1
      IPLT(II) = IGLAI
      IF (SGDD(I).GE.SDDAN .AND. KANTH.EQ.0) THEN
         IPLT(69) = IANTH
         KANTH = 1
      ENDIF
      IF (SGDD(I).GE.SDDPM .AND. KPMAT.EQ.0) THEN
```

```
         IPLT(69) = IPMAT
         KPMAT = 1
      ENDIF
 5860 IF (NOBS .EQ. 0) GO TO 5880
         DO 5870 IX=1, NOBS
         IF (IDAY .NE. IDATE(IX)) GO TO 5870
         II = IFIX(OBLAI(IX)*5. +.5) +1
         IPLT(II) = IOBS
 5870    CONTINUE
C     Plot line of graph for this day-
 5880 IF ((JDAP/10)*10 .EQ. JDAP) THEN
         WRITE(20,5890) JDAP,(IPLT(II), II=1,71)
 5890    FORMAT(1X,I3,1X,71A1)
      ELSE
         WRITE(20,5900) (IPLT(II), II=1,71)
 5900    FORMAT(5X,71A1)
      ENDIF
 5910 CONTINUE
      WRITE(20,5920)
 5920 FORMAT(5X,'0',8X,'1000',6X,'2000',6X,'3000',6X,'4000',
     2       6X,'5000',6X,'6000',6X,'7000'/
     3       26X,'ABOVE-GROUND DRY MASS (G/M2)')
C
 5930 WRITE(*,*) 'DO YOU WANT TO WRITE A LISTING OF THE DAILY'
      WRITE(*,*) 'SIMULATION RESULTS TO THE OUTPUT FILE ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,5000) ANS
      IF (ANS.EQ.NO1 .OR. ANS.EQ.NO2) GO TO 5970
      WRITE(20,5940)
 5940 FORMAT(///'LISTING OF DAILY SIMULATION RESULTS-'/
     2       5X,'DAY  DAP  DAE   SUM GDD     GLAI    ',
     3       '  AGDM   CUM AGDM'/)
      IEMER = 999
      KEMER = 0
      DO 5960 I=1, IEND
      IDAY = I + ISTRT -1
      IF (IDAY .LT. ISOW) GO TO 5960
      JDAP = IDAY -ISOW
      IF (SGDD(I).GE.SDDEM .AND. KEMER.EQ.0) THEN
         IEMER = IDAY
         KEMER = 1
      ENDIF
      JDAE = MAX0(IDAY-IEMER,0)
      IF (IDAY .GT. NYEAR) IDAY = IDAY -NYEAR
      WRITE(20,5950) IDAY,JDAP,JDAE,SGDD(I),GLAI(I),AGDM(I),TAGDM(I)
 5950 FORMAT(3X,3I5,F10.1,F9.2,F10.1,F11.1)
 5960 CONTINUE
C
 5970 WRITE(*,*) 'GRAMI IS FINISHED'
      STOP
      END
      SUBROUTINE REPLAC(P1,P2,P3,E1,E2,E3,P,E)
C     This subroutine includes the newest point and throws
C     out the oldest point in the bracketing search
```

```
      P1 = P2
      E1 = E2
      P2 = P3
      E2 = E3
      P3 = P
      E3 = E
      RETURN
      END
      SUBROUTINE REVERS(P1,P2,E1,E2,P,E)
C     This subroutine switches the order of the points as a
C     result of a reversal in the bracketing search direction
      P2 = P1
      E2 = E1
      P1 = P
      E1 = E
      RETURN
      END
      SUBROUTINE PARAB(P1,P2,P3,P4,E1,E2,E3)
C     This subroutine calculates the midpoint of the
C     parabola passing through the three points
      IF (P1 .GT. P3) THEN
         P  = P1
         P1 = P3
         P3 = P
         E  = E1
         E1 = E3
         E3 = E
      ENDIF
      XN = ( (P2-P1)**2*(E2-E3) - (P2-P3)**2*(E2-E1) )
      XD = 2.*( (P2-P1)*(E2-E3) - (P2-P3)*(E2-E1) )
      IF (XD .GT. 0.) XN = -XN
      XD = ABS(XD)
      P4 = P2 + XN/XD
      RETURN
      END
      SUBROUTINE RESEL3(P1,P2,P3,P4,E1,E2,E3,E4)
C     This subroutine selects three of the four points
C     to make up the new parabola
      IF (E4 .LE. E2) THEN
         IF (P2 .GE. P4) THEN
            P3 = P2
            E3 = E2
            P2 = P4
            E2 = E4
         ELSE
            P1 = P2
            E1 = E2
            P2 = P4
            E2 = E4
         ENDIF
      ELSE
         IF (P4 .GE. P2) THEN
            P3 = P4
            E3 = E4
```

```
    ELSE
       P1 = P4
       E1 = E4
    ENDIF
 ENDIF
 RETURN
 END
```

Model Execution

Execution of GRAMI is typically initiated by submitting the program executable file to the computer. Once this is done, the following message will appear on the terminal:

```
PC-Based Version 1.0 of gramineous crop model developed
by Stephan J. Maas at the USDA/ARS Subtropical Agric.
Res. Lab., Weslaco, Texas. Reference: Maas (1992) USDA
ARS-91.
```

SIMULATION SUMMARY WILL BE WRITTEN TO UNIT 20
File name missing or blank - please enter file name UNIT 20?

After this message appears, the user must key in a name for the output file that GRAMI will create to contain the results of the simulation. The last two lines of the message, concerning the missing or blank file name on UNIT 20, is the Microsoft compiler's response to the blank file name field in the OPEN statement for this I/O unit (program line 015). This response may be different if GRAMI is compiled using a different compiler. The response can be avoided by replacing the blank field in the OPEN statement with a file name prior to compiling the program.

Once the file name has been entered, GRAMI will ask:

DO YOU WANT TO WRITE A HEADLINE ON THIS SUMMARY?
(ENTER Y OR N)-

This question presents the user with an opportunity to write on the output file an introductory line of text that can be used to identify the simulation. If the user answers "yes," by entering an upper or lower case Y, the user is prompted to enter the contents of the headline (a maximum of 60 characters). If the user answers "no," by entering an upper or lower case N, a headline is not written on the output file.

The next response by GRAMI will be:

WEATHER DATA (MAX OF 200 DAYS) WILL BE READ FROM UNIT 10
File name missing or blank - please enter file name UNIT 10?

The user must key in the name of the existing file containing the daily temperature and irradiance data. Again, the last two lines of the statement result from the blank file field in the OPEN statement for I/O unit 10 (program line 030). The program requires file data to be in a specific format (see the Technical Description of Model section of this document). The utility program CONV_WX (see appendix B) can be used to extract the required information from an existing weather data file and to put it in the required format. The input file can contain a maximum of 200 days of data. This limit is determined by the array sizes for the environmental variables set in the DIMENSION statement (program line 003).

Once the file name has been entered, GRAMI will display the following message:

ENTER THE YEAR AND DAY OF YEAR OF PLANTING
(SEPARATED BY A COMMA)-

The model uses the year to determine whether planting occurred in a leap year. The day of the year (often incorrectly called the Julian day) is the sequential number of days from the start of the year. For example, if planting had occurred on December 31, 1984 (a leap year), the appropriate response to this message would be 1984, 366.

The next response by the model is as follows:

GROWTH OF THE FOLLOWING CROPS CAN BE SIMULATED-
1 Spring Wheat
2 Corn (Maize)
3 Grain Sorghum
4 A Different Crop
ENTER THE NO. OF THE CROP YOU WANT TO SIMULATE-
(NOTE- IF YOU ENTER NO. 4, YOU MUST INPUT THE PARAMETER DEFAULT VALUES FROM A DISK FILE)

By selecting a crop (entering 1, 2, or 3), the user determines which set of existing default parameter values will be used in the simulation. Built-in default values for crop numbers 1-3 are stored in a DATA statement (program lines 057-062). These values can be changed by editing and recompiling the program before executing GRAMI or by entering an entire set of default values from a disk file. If crop number 1, 2, or 3 is selected, GRAMI will ask:

DO YOU WANT TO ENTER DEFAULT PARAMETER VALUES FROM A DISK FILE ?
(ENTER Y OR N)-

If the user answers "yes," the user will be prompted to key in the name of the disk file. The model requires the data in the disk file to be in a specific format; an example of such a file is presented in appendix D. If the user answers "no," the appropriate built-in default values will be used.

If the user selects crop number 4, the parameter default values must be entered using a disk file. These values will replace the old default values for crop number 1 but are in effect only for the current simulation.

The model will then generate the statement:

MODEL PARAMETER DEFAULT VALUES FOR THIS CROP ARE-

The statement will be followed by a listing of 20 parameters and their default values that were obtained for the selected crop from the disk file or built-in array. The user will then be asked:

DO YOU WANT TO CHANGE A DEFAULT VALUE ? (ENTER Y OR N)-

By answering "yes," the user will have the opportunity to change the value of individual parameters in the list and will thus be prompted by the statement:

ENTER THE DEFAULT NO. (1-20) AND NEW VALUE (SEPARATED BY A COMMA)-

The appropriate response to change the default value for parameter 6 (light extinction coefficient) to 0.45, for example, would be 6,0.45. When a default value is changed, the list is redisplayed and the user will be asked if another default value should be changed. All changes are in effect for the current simulation only; they do not affect the default values stored in the disk file or built-in array.

Once the parameter default values have been specified, the model will ask:

DO YOU WANT TO INPUT OBSERVED GLAI VALUES ?
(ENTER Y OR N)-

Answering "yes" does not commit GRAMI to using the numerical solution technique for the simulation. Instead, the user may have GRAMI produce a growth simulation in which all parameters are held constant at their default values. The resulting output can be compared by the user to the observed values using the simple plotting routine contained in GRAMI. However, answering "no" to this question precludes using the numerical solution technique, since it relies on observed values of GLAI.

If the user answers "yes," the model will ask:

HOW DO YOU WANT TO INPUT THEM ?
(ENTER 1 FOR DISK FILE, 2 FOR KEYBOARD)-

"Them" refers to the observed GLAI values. It is relatively easy to input a small number of GLAI values from the keyboard. However, since inputted GLAI values are in effect for only one simulation, file input is easier for a large set of observations or for a small set of observations that will be used in a number of separate simulations. If the user chooses to input the GLAI values from a file, the program will ask the user to key in the name of the existing disk file containing the observations. The model requires this file to be in a specific format (see the Technical Description of Model section of this document) and the GLAI observations to be in chronological order. The utility program CONV_OB (appendix C) can be used to extract the required information from an existing data file and to arrange it chronologically in the proper format. GRAMI will read a maximum of 50 GLAI values from the disk file. This limit is set by the array size defined for this variable (OBLAI) in a DIMENSION statement (program line 003).

If the user chooses to input the GLAI values from the keyboard,
GRAMI will generate the statement:

ENTER NUMBER OF GLAI OBSERVATIONS (MAX OF 50)-

A maximum of 50 values can be input. Once the appropriate number has been entered, the program will prompt the user for the observation date and corresponding observed GLAI value for each sequential observation:

ENTER DAY OF YEAR AND GLAI VALUE (SEPARATED BY A COMMA)
FOR OBSERVATION NO. *n* -

Here, n is the number of the observation. If a GLAI value of 2.5 was observed on March 3, 1984 (a leap year), for example, the appropriate response to this statement would be 63,2.5. Observations must be entered in chronological order.

After all the observed GLAI values have been input, the model will ask:

SHOULD NUMERICAL SOLUTION TECHNIQUE BE USED ?
(ENTER Y OR N)-

Answering "yes" commits GRAMI to using the numerical solution technique for this simulation. In this case, the program will generate the following statements:

THERE ARE 5 FORMS OF THE NUMERICAL MODEL-
Form 0- No numerical solution used
Form 1- Simple solution (only XGLAI manipulated)
Form 2- XGLAI and parameter C manipulated
Form 3- XGLAI and parameters B and C manipulated
Form 4- Full solution (XGLAI, A, B and C manipulated)
PRELIMINARY EXAMINATION OF THE GLAI OBSERVATIONS SUGGESTS THAT MODEL FORM *m* SHOULD BE USED. DO YOU WANT TO USE A DIFFERENT FORM THAN THIS ?
(ENTER Y OR N)-

Here, m is the number of the model form identified by GRAMI as most appropriate for fitting the GLAI simulation to the GLAI observations (see the Technical Description of Model in this document). If the user enters "yes," the user will be asked to enter the number of the desired model form. If the user enters "no," the default form selected by GRAMI will be used.

The model is now ready to simulate the daily growth of the crop. If the numerical solution technique was chosen, the following message will appear:

PRESS Enter KEY TO START ITERATIVE NUMERICAL SOLUTION

Once the Enter key is pressed, the terminal will display the results of each iteration. On some machines, this information will pass across the terminal display too fast for the user to read. Prior to pressing the Enter key, the user may want to route the terminal display of the simulation to another device (such as a line printer) that will provide a permanent record for later study. A printout can be obtained on a PC by pressing the control (Ctrl) and print screen (Prt Sc) keys at the same time.

After the numerical solution technique has started, a pause may occur, and the following message will appear:

TOTAL ERROR (x) IS LESS THAN TOLERANCE (y)-
DO YOU WANT TO ACCEPT THIS AS THE SOLUTION ?
(ENTER Y OR N)-

Here, x is the total error for the current iteration, and y is the tolerable error specified by the user. This message indicates that the simulated GLAI values are relatively close to the corresponding observed values, although the convergence criteria for all parameters have not been met. The user has the opportunity to accept this iteration as the solution by

answering "yes" to the question. If the user answers "yes," the numerical solution stops. If the user answers "no," the numerical solution continues (although this message may reappear in succeeding iterations).

When the iterative numerical solution technique is finished, the following statement will be displayed:

ITERATIVE NUMERICAL SOLUTION IS FINISHED

This statement is not displayed if the user did not use the numerical solution technique. In either case, the user will be asked:

DO YOU WANT TO PRODUCE A ROUGH GRAPH OF THE MODEL SIMULATION WRITTEN TO THE OUTPUT FILE ?
(ENTER Y OR N)-

This graph is useful to quickly evaluate the general performance of the model (see the Technical Description of Model section for a description of this graph). A more detailed evaluation can be performed by answering "yes" to the next question asked by the model:

DO YOU WANT TO WRITE A LISTING OF THE DAILY SIMULATION RESULTS TO THE OUTPUT FILE ?
(ENTER Y OR N)-

This listing contains values of growing degree-days, GLAI, and aboveground dry mass computed for each day of the simulation. If the numerical solution technique was used, these values will come from the last iteration of the numerical solution. This data listing can be extracted from the output file and can then be analyzed using independent statistical and plotting programs.

The completion of the model simulation is signaled by the statement:

GRAMI IS FINISHED

The user should bear in mind that iterative numerical techniques do not always converge on a solution. The numerical techniques built into GRAMI have been designed to be relatively stable. Under certain conditions, however, the solution may not converge. This will be evident by the poor fit of the GLAI simulation to the observations or by the generation of a fatal error that abnormally terminates the simulation. Failure to converge typically results when one or more parameters used in the numerical solution are assigned default values that are inordinately different from the values required for convergence. The problem can be solved by specifying more appropriate parameter default values. Techniques for estimating parameter default values are presented in the following section of this document.

Technical Description of the Model

GRAMI is essentially a simple crop growth simulation model contained within an inner shell of numerical solution techniques and an outer shell of input/output operations. This section describes the details of how each part works to produce a simulation of crop growth and yield.

Crop Growth

Modeling daily crop growth over the growing season involves quantifying three basic activities in the following order: (1) absorption of photosynthetically active radiation (PAR) based on the amount of leaf area in the crop canopy, (2) production of new aboveground dry mass based on absorbed PAR, and (3) appearance or disappearance of leaf area associated with the production or loss (senescence) of dry mass. GRAMI evaluates these growth activities on a time schedule that is based on accumulated growing degree-days (GDDs) since planting.

Model growth activities are contained in the daily loop commencing in program line 384. This loop is traversed once each day, starting with the first day for which there are temperature and PAR data (ISTRT). The variable IDAY, evaluated in program line 385, is the current day of the year in the simulation. To simplify timekeeping, GRAMI continuously increases IDAY over the duration of the simulation, even when the simulation spans consecutive years. For example, if the simulation started on day 350 of the year 1984, the value of IDAY on January 10, 1985, would be 376. The user is typically unaware of this feature because dates are converted to or from this continuously increasing form when they are input or output by the model. The variable I appearing in the daily loop is the index used to locate arrayed data associated with IDAY. In program line 384 NDAYS is the number of days of environmental data in the simulation.

State Variables. GRAMI contains four state variables: sum of growing degree-days (SGDD), green leaf area index (GLAI), aboveground dry mass (AGDM), and total aboveground dry mass (TAGDM). Early in the growing season, AGDM and TAGDM are essentially the same. Later in the growing season, AGDM is different from TAGDM in that AGDM does not contain the leaf dry mass that has since disappeared from the crop canopy through senescence. State variable simulated values are saved in arrays for each day so that plotting and listing results will be facilitated at the end of the simulation. Since the state variables must be evaluated daily, they are assigned their respective initial conditions (program lines 387-390) at the start of each day of the simulation. All of the state variables will remain at their initial values (unchanged) if the model determines that IDAY is before planting (ISOW). If IDAY is after planting but before crop emergence, the values of GLAI, AGDM, and TAGDM will remain at their initial values.

Since the accumulation of GDDs starts at planting, the initial condition for SGDD is zero. The initial condition for GLAI (XGLAI) is small but must be greater than zero for simulated growth to occur. XGLAI should be roughly equivalent to the product of the plant population (plants m^{-2}) and the average area (m^2) of the first leaf on the crop plants. If the numerical solution technique is used, the model will start with the default value of XGLAI supplied by the user and iteratively compute a value of XGLAI that results in convergence of the numerical solution. This process is described later in this document. The initial condition for AGDM and TAGDM (XAGDM) is determined in program line 379 by dividing XGLAI by the specific leaf area (SLA, $m^2\ g^{-1}$). SLA is the ratio of unit area to unit mass of the leaf tissue produced on the current day. This parameter will be described in more detail later in this section.

Program line 392 determines whether the current day in the simulation (IDAY) is after the planting date (ISOW). If so, the state variable SGDD is increased in the next program line by the GDDs determined for IDAY. In GRAMI, GDDs are computed as the difference between the average daily temperature (TAVE) and the base temperature (BASET) specified by the user. The intrinsic function AMAX1 ensures that if TAVE is less than BASET, the value of SGDD will be increased by zero. The value of BASET varies according to crop type. It is typically in the range of 7-12 °C for the warm-season gramineous crops (corn and sorghum) or 0-5 °C for the cool-season gramineous crops (cereals).

Estimate of Occurrence of Phenological Stages. GRAMI uses SGDD to determine the occurrence of phenological stages of growth. The three cardinal growth stages in GRAMI are emergence, anthesis (start of grain growth), and maturity (end of grain growth). It is assumed in the model that each of these stages occurs on a single day, although in the field each stage may actually last several days. The user must supply values for the sums of GDDs from planting to emergence (SDDEM), to anthesis (SDDAN), and to maturity (SDDPM). These values are likely to be dependent on the location and planting date of the field but can be estimated from published crop performance trials and historical weather records.

For some gramineous crops, such as corn and sorghum, it has been noted that anthesis often occurs shortly after the attainment of maximum LAI. This phenomenon was used in an early version of GRAMI (Maas 1987) to estimate the occurrence of anthesis based on the numerical fit of simulated to observed GLAI. This procedure is not employed in the current version of GRAMI, since there is evidence that crop canopy growth may not be strongly coupled to flowering for some gramineous crops. For example, similar distributions of LAI over time were observed for spring and winter wheat grown concurrently at Weslaco, TX (figure 1). However, anthesis was observed 103 days after planting the spring wheat and 133 days after planting the winter wheat. Future studies may explain the

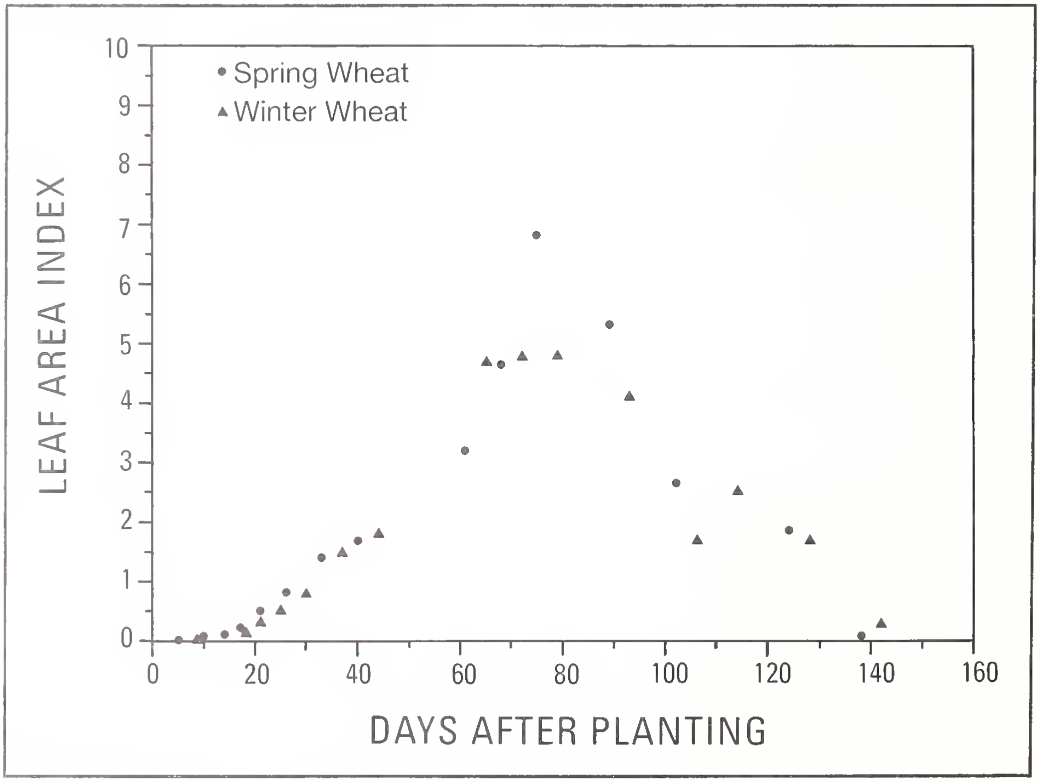


Figure 1
Measurements of leaf area index for spring and winter wheat grown at Weslaco, TX.

general relationship between flowering and crop canopy growth that can be incorporated into models such as GRAMI.

In the model, program line 394 determines whether the current day (IDAY) is after emergence. If so, the next program line determines if IDAY is after maturity. If this is the case, program lines 396-402 determine if the simulation should be terminated. If the numerical solution technique was not used (form zero was chosen), the simulation will be terminated at this point. Control will be transferred by line 398 to the end of the program, where results can be output. However, if the numerical solution technique was used, there may still be some GLAI observations after maturity to be considered in the numerical solution. In this case, the simulation will continue until after the day of the last GLAI observation. After the last observation is considered, control will be transferred by program line 401 to the numerical solution technique to determine whether or not convergence has been achieved. In either case, the variable IEND is evaluated to indicate the array index associated with maturity of the simulated crop. This variable is used later in plotting and listing model results.

Algorithms for Quantifying Growth. The algorithms in GRAMI that quantify the daily growth activities from emergence to maturity are contained between program lines 404-430. Daily absorption of PAR (APAR, MJ m^{-2}) is computed in program line 405. An exponential relationship, like that commonly proposed for crop canopies (Charles-Edwards 1982, p. 51, Charles-Edwards et al. 1986, p. 26), is used to relate APAR to the LAI established for the crop at the end of the previous day's growth. The parameter EX in this relationship is the "extinction coefficient," which represents a characteristic of the particular crop being modeled. A wide range of values for EX have been reported. However, after surveying the literature and reviewing data from my work, I believe that the values for EX are stratified according to crop type. EX values for the larger gramineous crops (corn and sorghum) tend to be in the range of 0.3-0.5. EX values for the smaller gramineous crops (cereals) tend to be in the range of 0.5-0.8. Differences in EX values are related to the leaf display characteristics of the various crops.

The calculation of daily dry mass production is performed in program line 407. Here, the daily increase in aboveground dry mass (DAGDM, g m^{-2}) is related to APAR using the parameter EF, a "conversion ratio" that empirically describes the efficiency with which absorbed solar energy powers carbon assimilation. As with EX, a wide range of values have been determined for EF, but the values of EF appear to be stratified according to crop type. EF values for the warm-season gramineous crops (corn and sorghum) tend to be in the range of 3.0-4.5 g MJ^{-1}. EF values for the cool-season gramineous crops (cereals) tend to be in the range of 1.5-3.0 g MJ^{-1}. Differences in EF values are related to the type of photosynthetic pathway (C_4 versus C_3) present in the crop. The total dry mass accumulation (TAGDM) is incremented by the current day's production (DAGDM) in program line 408.

Early in the growth of gramineous crops, most of the aboveground dry mass consists of leaf tissue. As the growing season progresses, however, the percentage of nonleaf tissues (stem and reproductive organs) in the aboveground dry mass increases. At some point, typically prior to flowering, the production of new leaf tissue ceases. In GRAMI, this phenomenon is quantified by the dimensionless parameter SPF (stem partitioning fraction). Program line 410 shows that following emergence, SPF increases from zero in an exponential manner as a function of GDDs. When SPF<1, the fraction of DAGDM partitioned to new leaf tissue is 1-SPF and the daily increase in GLAI (DGLAI) associated with the new leaf tissue is determined (program line 412). Specific leaf area (SLA) is used to relate the surface area of one side of the new leaf tissue to its mass. When the

computed value of SPF equals or exceeds 1, no new leaf growth is simulated and DGLAI will equal zero (program line 414). The curvature of the SPF function and hence the production of new leaf area are controlled by the parameters A and B. Thus, parameters A and B have been included in the numerical solution technique. Default values for A and B must be supplied by the user for the specific crop to be simulated. An easy method for determining these default values will be described later in this section.

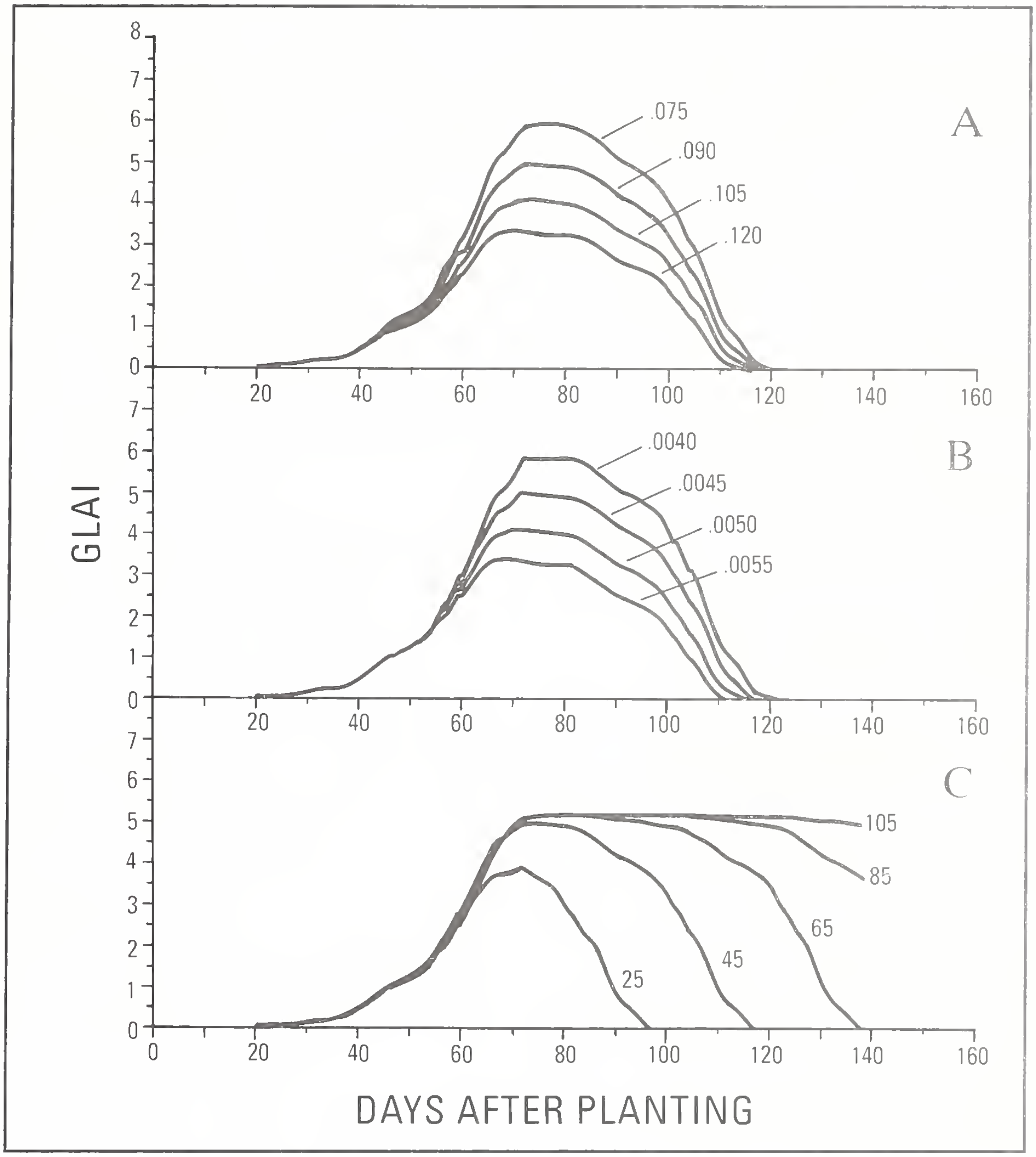
In the current version of GRAMI, the value of SLA is considered a constant. In reality, specific leaf area varies in relation to plant age, irradiance, and temperature (Friend 1966, pp. 193-195). Under constant temperature conditions and at irradiance levels commonly present in the field, the SLA of new leaf tissue tends to decrease with increasing plant age. This is due to an increase in the thickness of new leaves appearing on older plants. However, thickness of new leaves tends to decrease with increasing temperature. Thus, as the temperature increases over the growing season, these effects on leaf thickness and on SLA may to some degree cancel. Under most field conditions, values of SLA for gramineous crops tend to be in the range of 0.015-0.030 $m^2\ g^{-1}$

To calculate GLAI and AGDM for the current day, GRAMI considers not only the addition of new biomass from growth but also the loss of existing biomass from senescence. Senescence of leaf tissue is accounted for by estimating the expected lifespan (represented by parameter C, in days) of leaf tissue produced on a given day. The process for determining leaf tissue senescence is contained in program lines 417-426. In determining senescence, GRAMI sets the value of AGDM for the current day equal to TAGDM (program line 417) and initializes the value of GLAI for the current day as zero (program line 418). A loop (lines 419-426) is used to compare the age of leaf tissue produced on preceding days to the specified lifespan (C). The index of this loop (II) ranges from the start of the simulation to the current day (I). Days for which SGDD is less than SDDEM are not considered in the computation (program line 420), since no leaf tissue is produced prior to emergence of the crop. For each day following emergence, program line 421 computes the age of the leaf tissue produced on the respective day and compares it to the value of C. If the age is less than C, all leaf tissue produced on that day is considered to be still alive, and GLAI is increased in line 422 by the appropriate amount (DGLAI). However, if the age is greater than C, the leaf tissue produced on that day is considered to have senesced from the crop canopy. GLAI is not increased by DGLAI, and the dry mass associated with the senescent leaf tissue (computed from the quotient DGLAI/SLA) is subtracted from the accumulated aboveground dry mass (program line 424).

There is little in the literature concerning the general influence of plant age or environment on leaf lifespan. Jewiss (1966, p. 45) concluded that generally the rate of leaf senescence is about equal to the rate of leaf production in the *Gramineae*. However, Arkin et al. (1983) found that the interval (measured in GDDs) between the time when a leaf was fully expanded and the time when the leaf was half senesced increased for each leaf successively appearing on the stalk. Further investigation may resolve the effects of plant age on leaf lifespan. In the current version of GRAMI, the value of C is considered to be invariant over the growing season.

Manipulation of Parameters A, B, and C. During the development of GRAMI, it was found that manipulation of values for the parameters A, B, and C results in a variety of GLAI simulations that qualitatively resemble the growth of the various gramineous crops (figure 2). Thus, these three parameters were selected for use in the numerical solution technique. Parameters A, B, and C have been defined specifically only in the context of GRAMI and therefore are not found in the literature. The easiest way for the user to estimate default values for these parameters is to run the model as an analytical tool

Figure 2
Examples of GLAI simulations produced by GRAMI when a variety of values were used for parameter A, parameter B, and parameter C. The number associated with each curve represents the respective parameter value used in that simulatiion.



using a given set of GLAI observations. If an actual set of GLAI observations does not exist for the user's situation, the user can subjectively draw a curve representing the typical distribution of GLAI over the growing season based on the user's knowledge of crop growth in the region of interest. GLAI values can then be picked off this curve approximately every 10 days and used as input into GRAMI. A set of daily temperature and irradiance observations can be obtained from historical weather records, or temperature and irradiance data can be subjectively synthesized based on the user's knowledge of average conditions in the region. Built-in default values for the parameters BASET, EX, EF, and SLA can be used, or values can be specified from the ranges of values suggested earlier in this section. Specifying 100 GDDs as the default value for SDDEM should be sufficient for this analysis. The values specified for SDDAN and YF are not important in estimating default values for A, B, and C, so built-in defaults can be used. The default value for SDDPM can be determined by computing the sum of GDDs from the estimated date of planting to the estimated date of crop maturity using the weather data and BASET value selected for this analysis. Starting values for XGLAI, A, B, and C must be specified before starting the numerical solution. The accuracy of these starting values is not critical, since the objective of this analysis is for the numerical solution to ultimately converge on

the appropriate parameter values based on the set of GLAI observations provided by the user. Starting values may be determined as follows:

$$XGLAI = 0.01 \quad [1a]$$

$$C = DAYSPM/2 \quad [1b]$$

$$B = 0.003 \quad [1c]$$

$$A = e^{-B}\,(SDDMX) \quad [1d]$$

DAYSPM is the number of days between the estimated dates of planting and crop maturity. SDDMX (sum of GDDs from planting to the date of maximum GLAI) can be determined in the same manner as SDDPM. The built-in default values for the criteria that control the numerical solution technique (XDELA, XDELB, XDELC, CONVA, CONVB, CONVC, CONVG, and AVTOL) should be sufficient for this analysis.

Once all the information required for the user's first simulation has been collected, the program should be executed as described in the Model Execution section of this document. After the parameter starting values and GLAI observations are input, the numerical solution technique should be selected. When asked what model form should be used in the solution, the user should specify that form 4 (full solution) should be used. The values for A, B, C, and XGLAI produced in the first simulation should represent reasonable default values for use in later applications of the model.

Estimate of Grain Production. The remaining growth activity to be considered in GRAMI is the partitioning of dry mass to the grain. GRAMI assumes that a constant fraction (the value of parameter YF) of the daily increase in biomass (DAGDM) is contributed to the grain on each day between anthesis and maturity. The daily contribution to the simulated crop's grain yield is determined in program lines 428-429, where the variable YIELD is the mass of grain (g m^{-2}) on the current day. Analysis of data in the literature (Hanway and Russell 1969, Gibson and Schertz 1977) indicates that the value of YF for corn and sorghum is approximately 0.75. Values of YF for the cereals tend to be in the range of 0.9-1.2. Values of YF in excess of 1 indicate that carbohydrates produced in vegetative tissues prior to anthesis are remobilized during the grain filling period.

Numerical Solution Technique

The task of fitting the simulation of GLAI produced by GRAMI to observed values is a typical problem in the area of numerical analysis. The user is probably familiar with fitting a nonlinear curve, such as that produced by a polynomial equation, to observations. The polynomial equation, the initial guesses for the equation's parameters, and the observed values of the dependent variable of the equation are submitted to some nonlinear numerical technique that iteratively manipulates the parameter values until the curve produced by the equation "fits" (in the judgment of the user) the observations. The numerical solution technique in GRAMI does essentially this, with one difference. The GLAI simulation produced by GRAMI is not described by an equation—the simulation is a set of discrete values in which the dependent variable (GLAI) is not directly a function of the independent variable time (in days). Rather, the value of GLAI on any day is

determined by environmental conditions (temperature, irradiance) and by the state of the crop on the previous day. These characteristics of the simulation in part determined the type of numerical solution technique used in GRAMI.

Speed and stability were also considered in constructing GRAMI's numerical technique. Reasonably fast speed was desired so that the user wouldn't have to wait too long for a solution to converge. Unfortunately, features that enhance the speed of numerical techniques often decrease their stability. It was envisioned that GRAMI would be used by individuals who are more interested in evaluating the model results than in figuring out why the solution didn't converge. Thus, a numerical technique was selected that would have a high probability of eventually converging, even if the initial values for the parameters involved in the solution were not accurately specified.

GRAMI contains seven parameters (BASET, EX, EF, A, B, SLA, and C) and one initial condition (XGLAI) that can be manipulated to affect simulated growth. It is not necessary, however, for all of the parameters to be used in the numerical solution. Maas (1988a) showed that the portion of the difference between simulated and observed GLAI explained by each parameter decreases as additional parameters are included in the solution. A point may be reached where the inclusion of additional parameters does not markedly improve the accuracy of the simulation. This is particularly true when the model simulation is more sensitive to some parameters than to others. Optimum performance of the numerical technique should occur when the technique includes XGLAI, C, B, and A.

Computational "Layers." The numerical technique in GRAMI treats the portions of the solution involving XGLAI, C, B, and A as separate concentric "layers" of computation. The numerical solution progresses from the inner to the outer layer, i.e., convergence is first determined for XGLAI alone; then for XGLAI and C; then for XGLAI, C, and B; and finally, for XGLAI, C, B, and A. An advantage of this strategy is that it allows the solution to be halted at a layer below the outer layer. If the solution is halted, parameters in the outer layers will not be included in the solution. These constrained (or "degenerate") forms of the solution are used when the set of GLAI observations is too small to support a solution in which XGLAI, C, B, and A are all manipulated. A procedure has been included in GRAMI that, based on the GLAI observations provided by the user, will suggest what form of the solution should be used. This procedure will be described in detail later in this section.

When GRAMI is executed, the objective of the numerical solution technique is to place the GLAI simulation through the set of GLAI observations in the manner of a "best fit" curve. The simulation does not necessarily have to pass through each individual observation because the actual value of GLAI represented by each observation is not known precisely due to measurement error. A "best fit" of the GLAI simulation to the set of GLAI observations is achieved when the differences between the observed GLAI values and their corresponding simulated values are reduced to the minimal level allowed by the formulation of the model.

Figure 3 shows a sample plot of the differences (or "errors," if one views the simulation as an estimate of the observations) between some simulated and observed GLAI values. E_1, E_2, E_3, and E_4 represent the errors between observed and simulated values of GLAI in the plot. E_1 and E_2 are positive (above the simulated curve), while E_3 and E_4 are negative (below the simulated curve). EPOS is defined as the sum of the positive errors, while ENEG is defined as the absolute value of the sum of the negative errors. EPOS and ENEG provide the basis for determining how well the simulation fits the observations.

In GRAMI, EPOS and ENEG are evaluated in the Error Function Section (program lines 431-450) of the daily loop. For each day of the simulation, the current date (IDAY) is checked in program line 437 against the GLAI observation dates (IDATE). If an observation occurs on the current date, the error (ERROR) between observed and simulated GLAI values for that date will be computed in line 439. The contribution of ERROR values to either EPOS or ENEG is determined in lines 440-444. In program lines 376-377, EPOS and ENEG are set initially at zero before the start of the daily loop.

When the error of the last GLAI observation has been evaluated (in program line 446), the model can determine how well the simulation fits the set of GLAI observations. A goodness-of-fit statistic E is computed in line 448 by subtracting ENEG from EPOS. If E = 0, the simulation fits the observations such that the positive errors balance the negative errors. A simulation with E = 0 might be considered a "conditional" best fit because the total amount of error may not have been reduced to the minimal level. Nevertheless, the attainment of E = 0 is the first step in achieving the best fit.

Manipulation of XGLAI. It would be a great stroke of luck if the user were to specify default values for XGLAI, C, B, and A that would result in E = 0 on the initial pass through the model. It is more likely that the initial value of E will be greater or less than

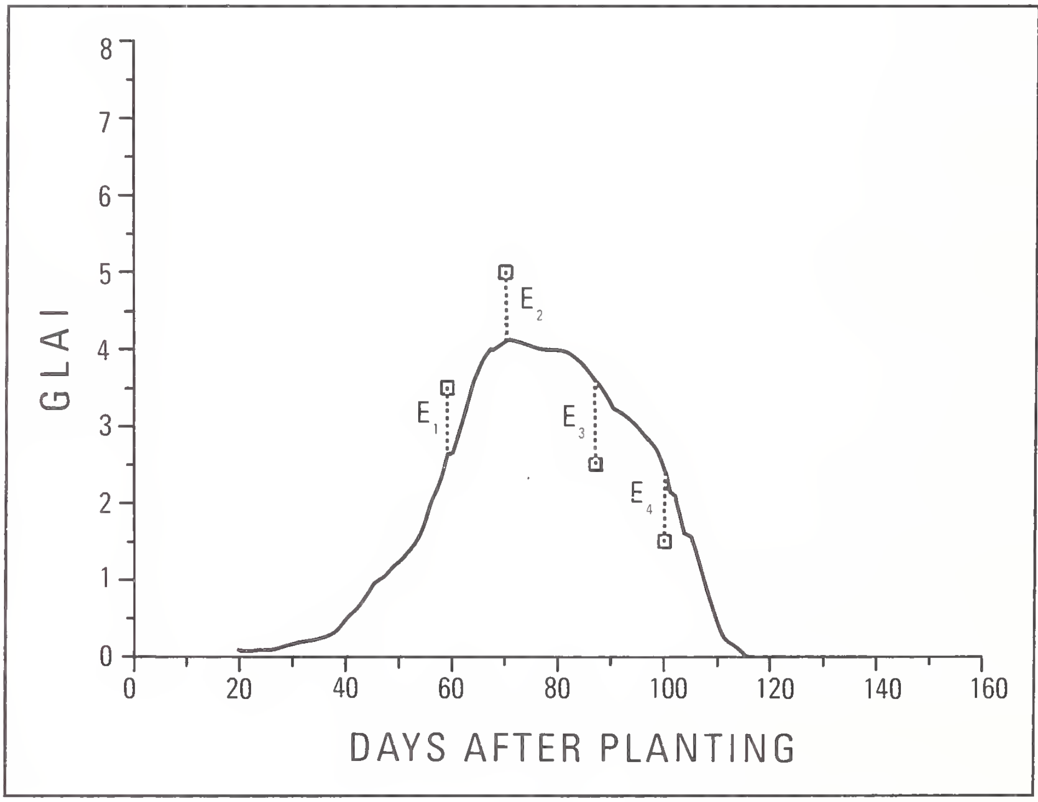


Figure 3
GLAI simulation (solid line) passing through four GLAI observations (squares). The dotted lines, denoted E_1 through E_4, represent the errors between observed and simulated values of GLAI. E_1 and E_2 are positive; E_3 and E_4 are negative.

zero. The numerical solution technique in GRAMI responds to a nonzero E by manipulating the value of XGLAI such that a conditional best fit is achieved. Maas (1988a, 1988b) showed that if all other model inputs, parameter values, and initial conditions are held constant, the value of E is a unique, nonlinear function of XGLAI for a given set of GLAI observations. An example of such an "error function" is shown in figure 4. The value of E at XGLAI = 0 is the sum of the observed GLAI values and will be positive as long as at least one GLAI observation is nonzero. Because the value of E continuously decreases as XGLAI increases from 0 to +∞, the error function will have one root (i.e., one XGLAI value that results in E = 0). Maas (1988a, 1988b) showed that this root can be determined by using an iterative technique such as the secant method. In the secant

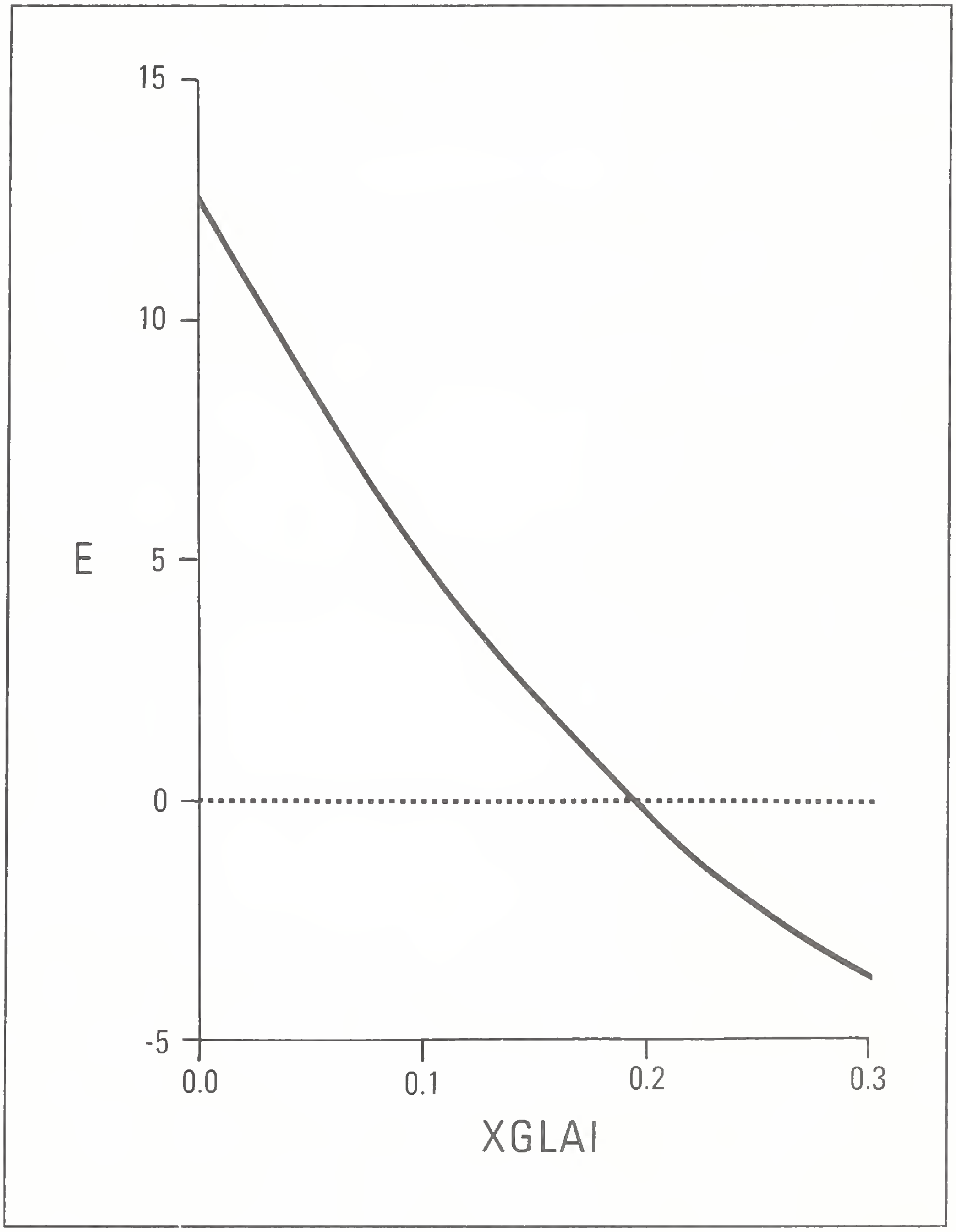


Figure 4
Error function (solid line) associated with simulations involving the GLAI observations in figure 3.

method, each iteration produces a complete growth simulation based on a given value of XGLAI. The method uses the XGLAI and E results of the previous and current iterations to determine the value of XGLAI for the next iteration:

$$(XGLAI)_{n+1} = (XGLAI)_n - (E)_n \left[\frac{(XGLAI)_n - (XGLAI)_{n-1}}{(E)_n - (E)_{n-1}} \right] \qquad [2]$$

where $(XGLAI)_{n+1}$ is the value for the next iteration, $(XGLAI)_n$ is the value for the current iteration, $(XGLAI)_{n-1}$ is the value used in the previous iteration, and $(E)_n$ and $(E)_{n-1}$ are, respectively, the values of the error function resulting from the use of $(XGLAI)_n$ and $(XGLAI)_{n-1}$ in the current and previous simulations. The bracketed term in equation 2 represents an approximation of the slope of the error function.

The iterative procedure starts when a simulation is made using the default value for XGLAI. The value of XGLAI for the second iteration is computed from the results of the first using one of the following equations:

$$(XGLAI)_{n+1} = (XGLAI)_n \quad [1 + (E)_n / EMAX] \qquad \text{if } (E)_n > 0 \qquad [3a]$$

$$(XGLAI)_{n+1} = (XGLAI)_n / \quad [1 - (E)_n / EMAX] \qquad \text{if } (E)_n < 0 \qquad [3b]$$

where $(XGLAI)_n$ is the default value for XGLAI, $(XGLAI)_{n+1}$ is the value of XGLAI for the second iteration, and EMAX is the value of the error function evaluated at XGLAI = 0. Once the initial and second iterations have been completed, equation 2 is used in succeeding iterations to rapidly converge on the root of the error function. Convergence is assumed to have occurred when the absolute value of $(E)_n$ is less than the user-specified value of CONVG.

The solution of the error function involving manipulating the value of XGLAI has been termed "reinitialization" and is performed in GRAMI in program lines 452-493. The test for convergence in the current iteration occurs in program line 456. If convergence has occurred, the current simulation and the reinitialization of XGLAI are considered to have been completed, and control is then transferred (program line 460) to the next layer of the numerical solution technique involving the parameter C. If convergence has not occurred, the model will prepare for another iteration. ITG, the number of the current iteration in the solution involving XGLAI, was set equal to zero (program line 375) for the initial iteration. For each succeeding iteration, ITG is increased by 1 (program line 466). The maximum number of iterations (MAXTRY) allowed by GRAMI within any layer of the numerical solution is 25 (program line 196). If line 467 indicates that ITG has exceeded MAXTRY, reinitialization will stop and the value of XGLAI from the last iteration will be accepted. In most cases, the solution will converge well before this limit. Program line 473 determines whether the new value of XGLAI will be determined using equation 2, 3a, or 3b. If ITG = 1, equation 3a or 3b will be used (program lines 477-481). The variables OLDE and OLDG, respectively, store the values of E and XGLAI from the previous iteration. If ITG > 1, equation 2 will be used (program lines 484-488). The variable SLOPE (line 484) represents the bracketed term in equation 2, and YINTC (line 485) is the new XGLAI value, temporarily stored, that will be used in the next iteration. The old E and old XGLAI values are saved as variables OLDE and OLDG, respectively (lines 486-487). It is possible (but unlikely) that the secant method may compute a negative value for XGLAI. A negative XGLAI is clearly unreasonable in the context of this

model, since XGLAI must be positive for the simulated crop to grow. If the secant method computes a negative XGLAI, program line 489 will replace the negative value with a positive value equal to OLDG/2, and the daily loop will restart (line 491), proceeding with the next iteration of the XGLAI solution.

Reinitialization represents the most constrained layer of the numerical solution, since the manipulation of XGLAI affects only the magnitude and not the shape of the GLAI simulation (Maas 1988a, 1988b). Although EPOS = ENEG following reinitialization, the individual terms EPOS and ENEG may still be large. The remaining layers of the numerical solution involve manipulating the parameters C, B, and A to change the shape of the GLAI simulation so that the sum of the individual errors (EPOS+ENEG) between simulated and observed GLAI is minimized.

Manipulation of Parameter C. The next layer of the numerical solution involves parameter C. The objective of this portion of the numerical solution is, while holding all other parameters at their default values, to find the values of XGLAI and C that result both in EPOS equalling ENEG and in minimizing EPOS+ENEG. To accomplish this objective, reinitialization is first performed with the value of C set at its default. Upon convergence, the value of the total error (EPOS+ENEG) is noted. Next, the value of C is decreased by a given amount, and reinitialization is performed using the new C value. Upon convergence, the total error associated with the new C value is compared with the total error associated with the previous (default) C value. If the total error has decreased, the search for the value of C that minimizes the total error is proceeding in the proper direction. If the total error has increased, then the search should proceed with values for C that increase, rather than decrease, from the default value. Thus, the first iteration identifies the "search direction." Additional iterations are performed until the value of the total error no longer decreases. The value of C that minimizes the total error is called the bracketed root. This value of C is the root of the portion of the numerical solution involving the parameter C and is bracketed by the C values of the two most recent iterations.

In some predecessors of GRAMI, the domain between the bracketing C values was repetitively searched with ever-decreasing differences between successive C values until total error was minimized. However, this process for finding the bracketed root was inefficient. A technique called "parabolic interpolation" (Press et al. 1986, p. 283) is used in the current version of GRAMI and leads to rapid convergence on the bracketed root. In this technique the shape of the error function involving the parameter C is assumed to resemble a parabola in the region near the root (figure 5). The abscissa value associated with the bottom (minimum) of this parabola corresponds with the C value that is associated with minimal total error. This C value can be computed if three points on the parabola are known. If the C values and total error values (EC) from the three most recent iterations are taken as points on the parabola, the C value at the bottom of the parabola is given by

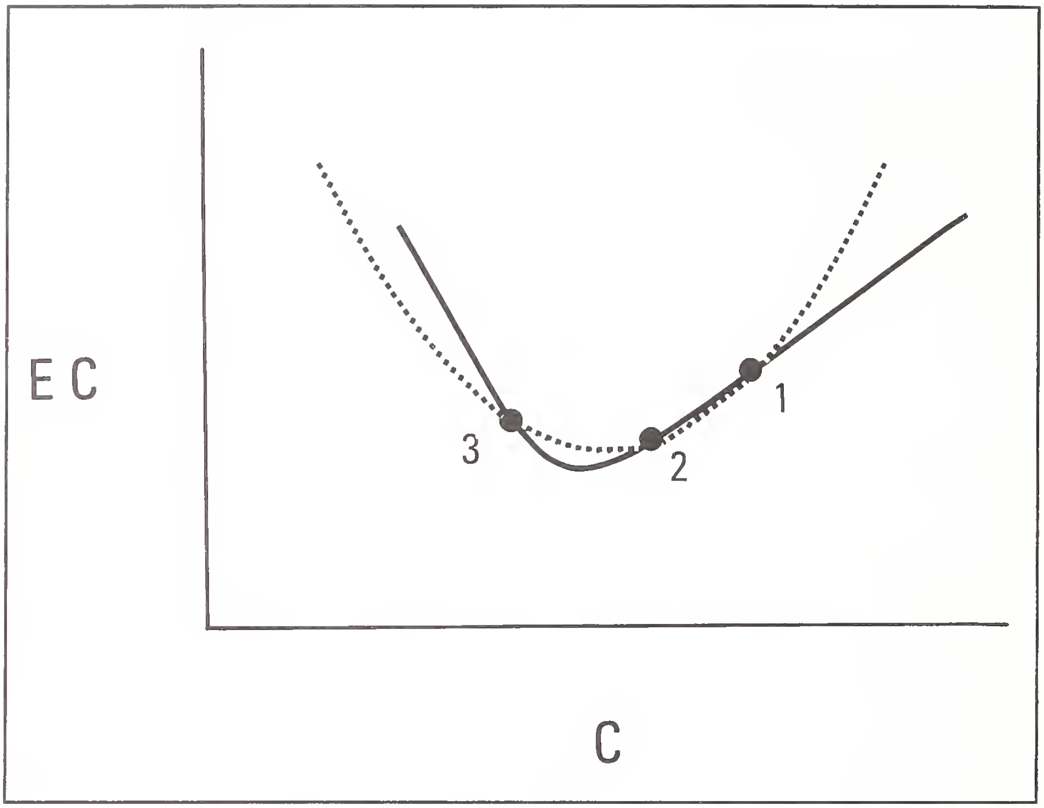
$$(C)_{n+1} = (C)_{n-1} + \frac{1}{2}\frac{(XN)}{(XD)} \qquad [4]$$

$$XN = [(C)_{n-1} - (C)_{n-2}]^2\,[(EC)_{n-1} - (EC)_n] - [(C)_{n-1} - (C)_n]^2\,[(EC)_{n-1} - (EC)_{n-2}]$$

$$XD = [(C)_{n-1} - (C)_{n-2}]\,[(EC)_{n-1} - (EC)_n] - [(C)_{n-1} - (C)_n]\,[(EC)_{n-1} - (EC)_{n-2}]$$

where $(C)_{n+1}$ is the value of C at the bottom of the parabola and the subscripts n, n-1, and n-2, respectively, apply to the most recent, the second most recent, and the third most

Figure 5
Parabola (dotted line) fit to three points on the error function (solid line) involving parameter C. EC is the total error associated with C.

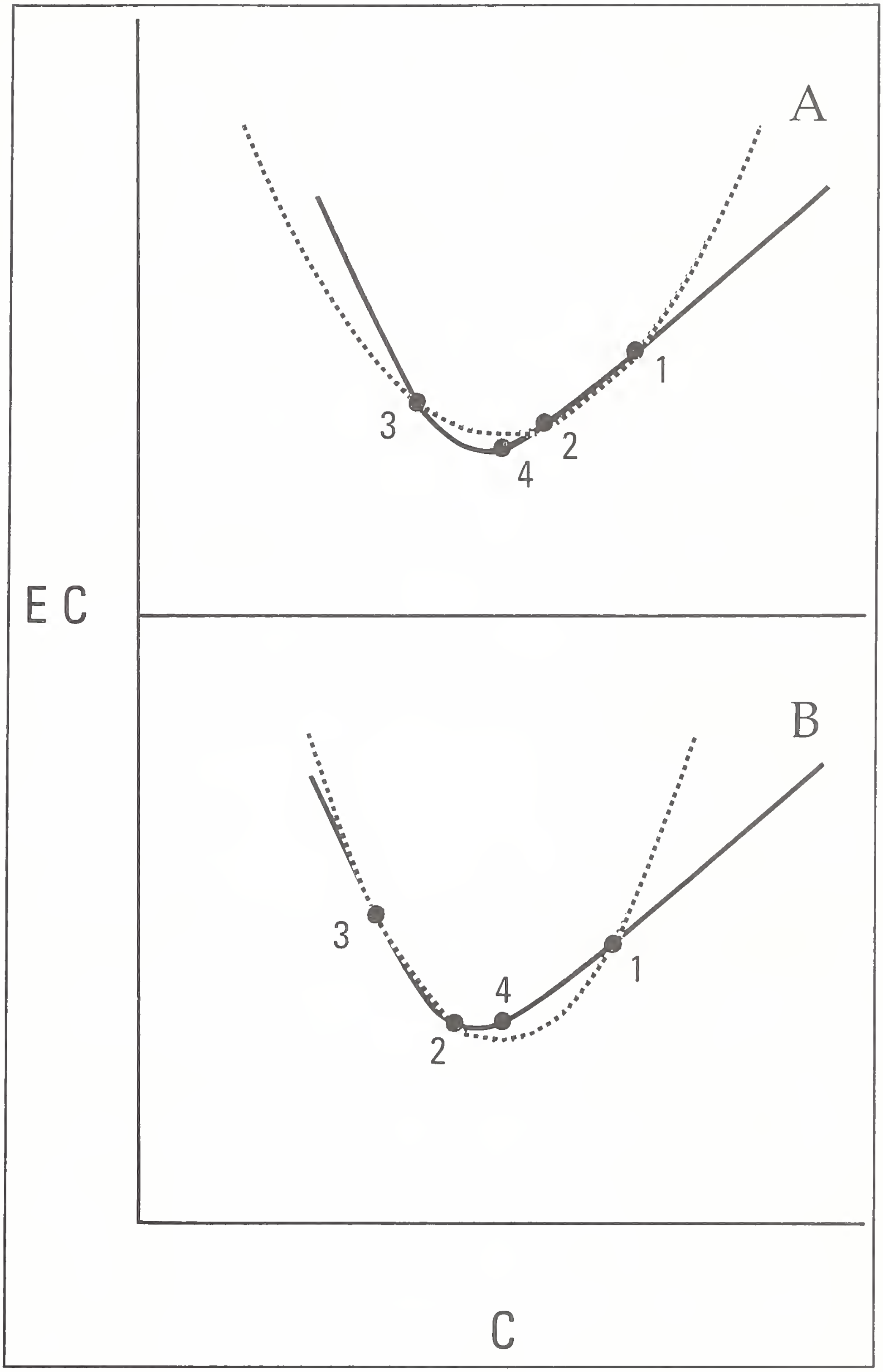


recent iterations involving C and EC. Convergence of the portion of the solution involving C is assumed if the absolute difference between $(C)_{n+1}$ and $(C)_{n-1}$ is less than the user-specified value of CONVC, and $(C)_{n+1}$ is considered to approximate the root of the error function involving the parameter C.

If convergence is not indicated, the iterative procedure continues. Reinitialization is performed using $(C)_{n+1}$, and the total error associated with the new value of C from this iteration is determined. The new value of C and two of the three previously established points on the error function may be used to approximate the minimum value of a new parabolic representation of the error function. Of the previously established points, $(C)_{n-1}$ and the other point that brackets $(C)_{n-1}$ and $(C)_{n+1}$ will be used for this computation (figure 6). This bracketing search procedure generally requires only a few iterations to converge on an approximation of the root.

The procedure that uses bracketing search and parabolic interpolation has been termed "reparameterization." Program lines in GRAMI that perform reparameterization for parameter C (lines 343-373 and lines 499-562) bracket the portion of the model that performs reinitialization, which is part of the daily loop that computes crop growth. Before reparameterization starts, the "step size" (DELC) used to determine the C value in succeeding iterations of the bracketing search is initially equal to the user-supplied default XDELC (program line 336). The number of the current iteration involving C (ITC) is set at zero (line 337). The value of the indicator variable KCPAB is also set at zero (line 338), indicating that the bracketing search portion of reparameterization is in effect. When the root is bracketed, the value of KCPAB will be changed to 1, indicating that reparameterization has switched from bracketing search to parabolic interpolation. For the first iteration (ITC = 0), the value of C equals its default, so line 339 transfers execution past the program lines that normally determine the value of C for the new iteration. In succeeding iterations, the value of ITC is increased by 1 in program line 343. The value

Figure 6
Examples of parabolic interpolation in which the new parabola (dotted line) will be fit to the three points with the lowest error. In (A) points 2, 3, and 4 will be used. In (B) points 1, 2, and 4 will be used.



of XGLAI from the previous iteration is saved as the variable CSTG in line 344; the use of CSTG will be described later in this section. As in reinitialization, the number of the current iteration is checked to determine whether it exceeds MAXTRY (program lines 345-350). The value of C for the current iteration is determined in lines 353-357. If the search direction is positive (value of C increases with each succeeding iteration), the C value is computed (line 354) by increasing the previous C value by DELC*ITC. ITC is used as an "accelerator" because it makes the bracketing steps progressively larger for succeeding iterations, effectively speeding up the solution. If the search direction is negative (value of C decreases with each succeeding iteration), decreasing the previous C value by DELC*ITC might result in negative C values (which would be meaningless in the context of this model). When the search direction is negative, line 356 will provide a value for C that is half the value of C in the previous iteration. Therefore, C will approach, but never equal, zero. Once C has been determined, it is tested in program lines 358-363 to determine whether it lies within the domain of C allowed by the model. This domain is bounded by maximum and minimum values (CMAX and CMIN, respectively) established in program lines 193-194. The purpose of CMAX and CMIN is to ensure that the value of C remains within the range that is meaningful in the context of the model. After the evaluation of C, program line 367 transfers control to the daily loop, which leads to the reinitialization segment.

When reinitialization has been completed, the total error (EC) associated with the current iteration involving C is determined in program line 500. Program lines 502-510 then determine whether the EC is less than the "tolerable" amount (TOLER) specified by the user. This comparison of EC with TOLER is included in GRAMI to prevent certain instances in which the numerical solution takes an inordinately long time to converge. A long period for convergence usually is the result of using a high-level form of the numerical solution technique (such as form 3 or 4) with a small number of GLAI observations (fewer than three). In such a case, the inner layers of the numerical solution rapidly bring the magnitude of the GLAI simulation into reasonable agreement with the observations. However, the outer layers of the numerical solution continuously alter the shape of the GLAI simulation in an attempt to bring the shape into finer agreement with the observations. When there are not enough GLAI observations to effectively constrain the shape of the simulation, this iterative process may continue for quite a long time. Thus, the model can determine whether the current simulation produced by the inner layers of the numerical solution is acceptable without waiting for convergence of the outer layers. The value of TOLER is computed in program line 195 as the product of the number of GLAI observations and the user-specified error (AVTOL) that is tolerable on average for each observation.

If the numerical solution continues and the root has not been bracketed, the test in program line 511 involving KCPAB allows the bracketing search to continue. If the current iteration is the initial one (ITC = 0), lines 513-517 will store the current values of C and EC as the variables CP1 and CE1, respectively, and will return the solution to the start of the reparameterization section for the next iteration. If ITC = 1, the search direction can be determined (lines 518-530). If the current value of the total error (EC) is less than the value from the previous iteration (CE1), then the current search direction (indicated by the sign of DELC) is appropriate. In this case, the current values of C and EC are saved as the variables CP2 and CE2, respectively, and the values from the previous iteration are saved as CP1 and CE1. If the test in program line 520 indicates that EC is greater than CE1, however, the search direction will be reversed. In this case, the current values of C and EC will be saved as CP1 and CE1, respectively, and the results of the previous iteration will be saved as CP2 and CE2. This switch in the ordering of CP and CE values is performed by the subroutine REVERS (program lines 817-825), which is called in line 524. Once the switch in the ordering has occurred, the numerical solution technique will not

return to the start of the reparameterization section; doing so would make the next iteration a duplicate of the initial iteration. This duplication is avoided by resetting the value of C to its initial value (program line 525) prior to returning to the start of the reparameterization section. The value of XGLAI does not have to be reset to its original default, since the value of XGLAI at convergence of the reinitialization procedure is known for the initial iteration involving C. The XGLAI value for the initial iteration was saved at the start of the current iteration as CSTG. XGLAI is reset to CSTG in program line 526. The search direction is reversed in line 527, prior to returning the solution to the start of the reparameterization section.

When ITC = 2, the values of C and EC from the current iteration are saved as CP3 and CE3, respectively (program lines 533-534). After three consecutive iterations of the numerical solution involving C, saved values of C and EC will include CP1, CE1, CP2, CE2, CP3, and CE3. In succeeding iterations (ITC > 2), the values of C and EC in the current iteration will be renamed and saved as CP3 and CE3, respectively. The values saved previously as CP3, CE3, CP2, and CE2 will be renamed, respectively, CP2, CE2, CP1, and CE1. The values saved previously as CP1 and CE1 will be thrown out. The renaming of C and EC values is performed by subroutine REPLAC (program lines 806-816) called from line 536. The test for bracketing the root occurs in line 538. If the root has been bracketed, the value of the indicator variable KCPAB is set at 1 (line 542), and the value of XGLAI is set equal to CSTG (line 543) at the start of the current iteration. The numerical solution then returns (line 549) to the start of the reparameterization section for the first iteration involving parabolic interpolation. If the root has not been bracketed by the current iteration, the solution is returned for the next iteration in the bracketing search.

Once bracketing has occurred, the test involving KCPAB in program line 351 allows the value of C to be determined using parabolic interpolation (lines 369-373). The new value of C is determined by subroutine PARAB (program lines 826-843) called by line 369. This subroutine evaluates equation 4 using the values saved from the three preceding iterations and returns the new value of C as the variable CP4. The value of C for the new iteration is set equal to CP4 in line 370, and the numerical solution proceeds to reinitialization.

Following completion of the reinitialization portion of the numerical solution, the test involving KCPAB in program line 511 allows activities associated with parabolic interpolation to continue (lines 551-562). The test in program line 551 determines whether convergence of the numerical solution involving C has occurred. If not, the current value of EC (fourth iteration) is saved as CE4 (the current value of C was saved earlier as CP4 by subroutine PARAB). The C and EC values from the current and three most recent iterations of the numerical solution are submitted in line 556 to subroutine RESEL3 (program lines 844-869). As described earlier, the purpose of this subroutine is to select three of the four sets of values to be used in determining the new parabolic approximation of the error function of C. Once these values are established, the solution is returned by line 557 to the start of the reparameterization section for the next parabolic interpolation iteration. If the test in line 551 indicates that convergence has occurred, reparameterization involving C is finished, and the value of EC represents the total error remaining after manipulation of XGLAI and parameter C.

Manipulation of Parameters B and A. The remaining layers of the numerical solution involve reparameterization for parameter B (lines 304-334 and 566-621) and parameter A (lines 265-295 and 625-682). The structure of these layers is identical to that just described for parameter C. Variable names in each section reflect the parameter that is

being manipulated (e.g., AP2 is the value of parameter A after the second iteration). The objective of reparameterization involving B is to further minimize the total error (EC) left after reparameterization involving C. The objective of reparameterization involving A is to further minimize the total error (EB) left after reparameterization involving B. The value of EA represents the total error left after reinitialization and after three layers of reparameterization (C, B, and A). Adding more layers to the numerical solution does not markedly reduce the remaining total error.

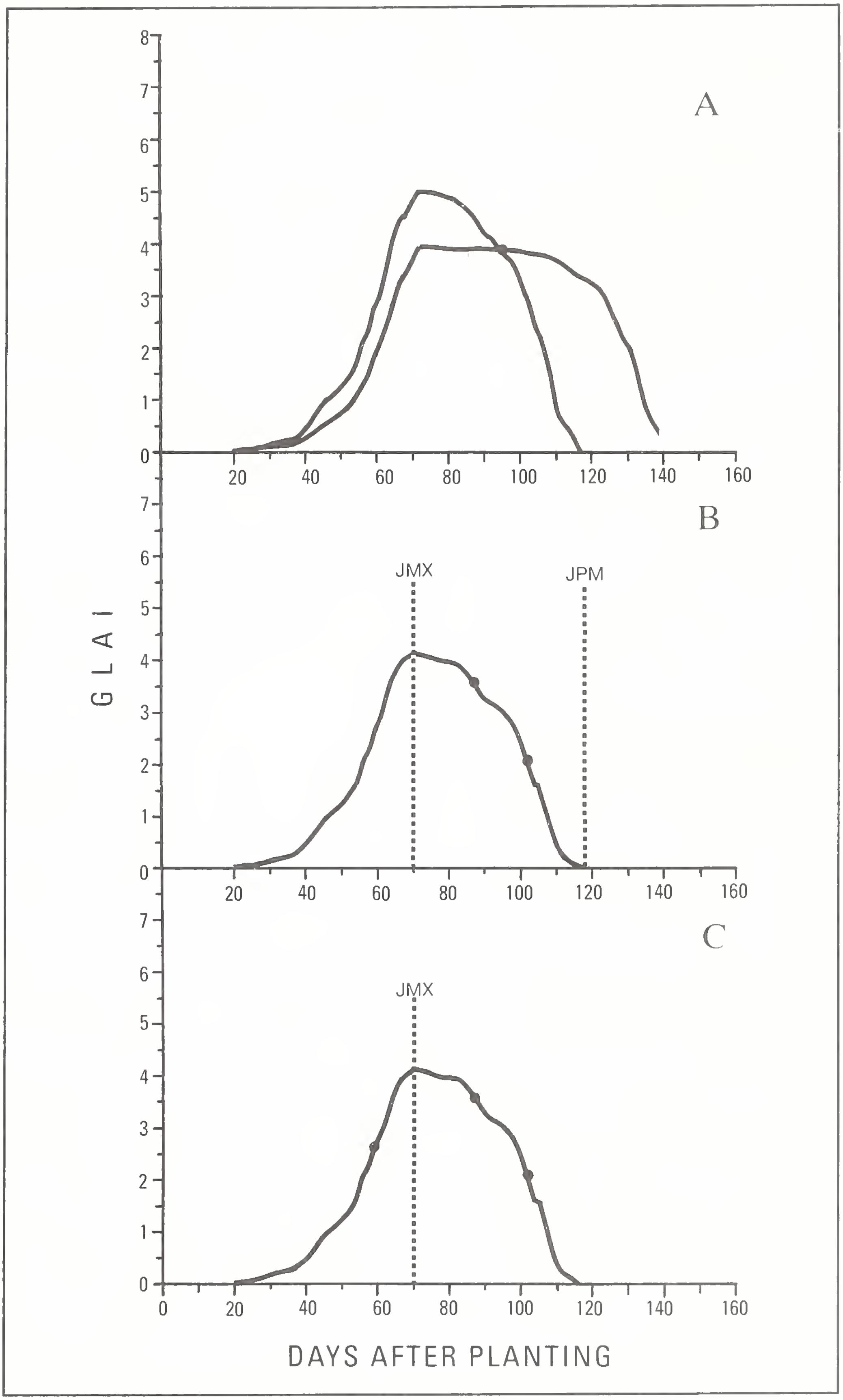
Determining Which Layers to Use. As mentioned earlier, the use of all layers of the numerical solution is not always appropriate for a given set of GLAI observations. Therefore, GRAMI's numerical technique was designed so that constrained forms of the technique can be used. After gaining some experience with the numerical technique, the user can probably guess which form of the solution is appropriate for use with a given set of GLAI observations. However, a procedure (program lines 199-253) has been included in GRAMI that will suggest, based on a cursory examination of the GLAI data, which form of the numerical solution might be most appropriate. The form of the numerical solution is indicated by the value (0 to 5) of the variable MODEL, which is initially set at zero (program line 148) before input of the GLAI observations. If the numerical solution technique is selected by the user but no GLAI observations are input (NOBS = 0), GRAMI (line 199) will maintain the value of MODEL at zero to indicate that the numerical solution technique cannot be used.

When GLAI observations are input, the form of the numerical solution suggested by GRAMI will be based on the number of observations and their occurrence relative to certain phenological events during the growing season. When only one GLAI observation is available during the growing season, an unlimited number of GLAI simulations could be fit through the observation by manipulating the shape of the GLAI simulation (two possibilities are shown in figure 7A). For such situations, reparameterization should not be used, since it involves the parameters that affect the shape of the GLAI simulation (C, B, and A). Only reinitialization (MODEL = 1), which affects the magnitude of the GLAI simulation through the value of XGLAI, is appropriate.

Higher order forms of the numerical solution might be appropriate if more than one GLAI value is input. The small loop (program lines 203-213) contains some of the relationships found in the larger daily loop. The purpose of the small loop is to estimate the dates of occurrence of maximum GLAI (JMX) and crop maturity (JPM) from the planting date, the average daily temperature, and the user-supplied default values for BASET, SDDEM, SDDPM, A, and B. The occurrence of the GLAI observations relative to JMX and JPM determines whether higher order forms of the numerical solution might be appropriate. Between JMX and JPM, GLAI typically decreases due to leaf senescence. As described in the previous section of this document, the rate of senescence is dependent on the value of parameter C. A minimum of two GLAI observations taken during the period between JMX and JPM (figure 7B) can indicate the general slope of the GLAI curve over this period and thus can be used to constrain reparameterization involving C (MODEL = 2). If the observations are close together, however, the variation between GLAI values may be due more to measurement error than to senescence. In this case, it would be better to treat the multiple GLAI values as though they were one observation. Thus, the model will not suggest that form 2 be used unless the distance (in days) between the first and last GLAI observations is at least one-third as great as the distance between JMX and JPM (determined in program lines 215-226). If this criterion is satisfied, the model will suggest that form 2 be used (program line 227).

The addition of at least one GLAI observation prior to JMX (figure 7C) can act to further constrain the shape of the simulation. The shape of the simulation prior to JMX is

Figure 7
Examples of GLAI simulations in which the number and timing of GLAI observations (circles) help determine which form of the numerical solution should be employed. Dashed vertical lines indicate the estimated times of maximum GLAI (JMX) and crop maturity (JPM).



related to the rate of leaf area production, which is determined by parameters B and A. Thus, if at least one GLAI observation occurs before JMX (in addition to two or more observations occurring between JMX and JPM), the model will suggest that the solution include reparameterization involving parameter B (MODEL = 3). This decision occurs in program lines 229-230. The option to use reparameterization involving parameter A (MODEL = 4) is left to the user. If there are enough observations to delineate the entire shape of the GLAI curve, then the use of the full solution (form 4) is probably appropriate.

The user ultimately decides which form of the numerical solution is to be used. Program lines 231-251 provide the suggested form of the numerical solution and allow the user to either accept the suggestion or specify an alternate form from among the five choices. Once the form is determined, the value of the variable MODEL will determine which portions of the numerical solution will be used and which will be skipped in the execution of the simulation.

Default Values. Prior to starting the simulation, the user may want to supply the default values for the criteria that control the numerical solution. These values include those for the step sizes (XDELC, XDELB, and XDELA) in the bracketing search technique, for the convergence criteria in the reparameterization (CONVC, CONVB, and CONVA) and the reinitialization (CONVG) procedures, and for the average tolerable error (AVTOL). For most situations, the default values provided in the built-in data statement (program lines 056-062) should be adequate for these criteria. Increasing the values of the step sizes may result in more rapid convergence of the numerical solution. However, if the step sizes are too large, the bracketing search process may not detect the error function minimum between successive iterations. Decreasing the values of the convergence criteria will generally result in a closer fit of the simulation to the observations. However, the numerical solution will then take longer to converge. Increasing the value of AVTOL will hasten convergence but will produce a less-accurate simulation.

Status of the Numerical Solution. Once the numerical solution technique has started, the current status of the numerical solution is written to the terminal display. The results of each iteration are presented so that the user can follow the unfolding of the solution on the computer screen (assuming that the output doesn't pass across the screen too fast to be read). An example of the output produced after starting a simulation is shown below:

```
A ITERATION      0, A = .038500 (BRACKETING SEARCH)
   B ITERATION   0, B = .003000 (BRACKETING SEARCH)
     C ITERATION   0, C = 50.50 (BRACKETING SEARCH)
       XGLAI  ITERATION 0, XGLAI = .250000000
       SOLUTION NOT CONVERGED (EPOS = 18.44, ENEG = 18.54)
       XGLAI ITERATION 1, XGLAI = .249175000
       SOLUTION NOT CONVERGED (EPOS = 18.46, ENEG = 18.53)
       XGLAI  ITERATION 2, XGLAI = .247062800
       SOLUTION CONVERGED (EPOS = 18.49, ENEG = 18.49)

     C ITERATION   1, C = 54.50 (BRACKETING SEARCH)
       XGLAI ITERATION 0, XGLAI = .247062800
       SOLUTION NOT CONVERGED (EPOS = 18.49, ENEG = 20.14)
                         •
                         •
                         •
```

```
      XGLAI ITERATION 2, XGLAI =1.410353000
      SOLUTION CONVERGED (EPOS = 10.63, ENEG = 10.63)

    C SOLUTION CONVERGED FOR C = 18.67  (EC = 21.260)

  B ITERATION   1, B =  .003100 (BRACKETING SEARCH)
    C ITERATION   0, C =  18.67 (BRACKETING SEARCH)
      XGLAI ITERATION 0, XGLAI =1.410353000
      SOLUTION NOT CONVERGED (EPOS = 10.74, ENEG = 10.17)
                        •
                        •
                        •
```

Note that some of the lines in the output are indented to distinguish the output associated with each layer of the solution. When a constrained form of the solution is used, the lines associated with the parameters that are not involved in the solution are omitted from the output. Two lines are written for most iterations of the solution. The first line identifies the parameter (A, B, or C) or initial condition (XGLAI) being manipulated, the number of the iteration, and the value specified for the parameter or initial condition in the iteration. For iterations involving A, B, or C, the reparameterization method (bracketing search or parabolic interpolation) is also identified in the first output line. The second line describes whether the solution converged or was bracketed and presents the resulting error (total error for reparameterization or EPOS and ENEG for reinitialization). Since three iterations are required to attain convergence or bracketing in the reparameterization technique, the second output line is omitted for iterations 0 and 1 involving parameters A, B, or C. A blank line is written to the screen following the convergence or bracketing of any portion of the solution so that these events may be more easily identified in the output.

Input/Ouput

The input/output (I/O) segments of GRAMI bracket the numerical solution technique. Input statements supply the information required to perform a simulation of crop growth; output statements inform the user of the simulation results. GRAMI has been designed so that I/O activities are uncomplicated, yet are flexible because they allow the user several options for entering data or viewing results. Input of information can occur from a keyboard or a disk file. Unless redirected to another I/O unit, the status of the model's performance is output to the terminal display. Results of the simulation are output to a disk file so that they may be studied in detail later.

I/O activities in the first portion of the model (program lines 014-186) input and organize information in preparation for the simulation of crop growth. GRAMI's sequence of requests for information from the user was described earlier in the Model Execution section of this document. These requests ask the user for yes/no decisions, for specific values, or for the name of a disk file (to be read from or written to). The answer to a yes/no question is read as the single-character variable ANS (program lines 018-019). The test of the contents of ANS against built-in indicator variables (YES1, YES2, NO1, and NO2, defined in line 005) determines whether the question has been answered in the affirmative or negative. Since ANS is a single-character variable, the responses "Y," "y," "YES," "Yes," and "yes" are acceptable for the affirmative; and "N," "n," "NO," "No," and "no" are acceptable for the negative. The test is typically the first line in an IF-THEN-ELSE-ENDIF structure (as in lines 064-071), which will direct GRAMI's action in response to the user's answer.

Requests for specific values (e.g., program lines 037-038) are made by GRAMI to obtain small amounts of information from the user. These values are read by GRAMI in free-format from the keyboard (line 039). When more than one value is requested by GRAMI, the user is informed to separate the values with a comma (a blank space can also be used). It is not necessary for the user to input a decimal point in whole-number values that will be read from the keyboard as real variables. The use of a decimal point in values being read as integers (such as dates) will result in the generation of an error message.

Requests for the names of disk files (e.g., program lines 028-029) are made by GRAMI to allow the user to input or output large amounts of information. A statement is written to the terminal display telling the user what I/O unit number will be associated with the disk file. An OPEN statement (program line 030) immediately follows each disk file request. If the program was compiled with a Microsoft FORTRAN Compiler (Version 4.0), the blank file field in the OPEN statement will generate a request to the user to input a disk file name. If another compiler was used, it may be necessary for the user to replace the blank field with a file name before compiling the program. The activities associated with each I/O unit accessed by the program are as follows:

Unit 10: Input of daily weather data
Unit 20: Output of simulation results
Unit 30: Input of parameter default values
Unit 40: Input of observed GLAI values

Input of Daily Weather Data. Daily weather data are input as directed in program lines 031-035. GRAMI requires these inputs to be in the following order and units: current day of the year (IDAY), average daily air temperature (TAVE, in °C), and photosynthetically active radiation (PAR, in MJ m^{-2}). Each input value must be separated by a blank space. If the user's weather data set has daily temperature and irradiance in a different format or in different units, the utility program CONV_WX (appendix B) can be used to convert the data into the required form. When executed, CONV_WX is self-explanatory and thus will not be explained here. Two additional variables, NDAYS (number of days of weather data) and ISTRT (day of the year on which the weather data starts), are evaluated after the weather data are input.

Output of Simulation Results. The output file for the simulation results is designed to contain all the information necessary to characterize the simulation. A headline containing a maximum of 60 characters may be written (program lines 016-026) at the start of the summary to identify the simulation. The values of ISTRT and ISOW (planting date) are written to the output file by lines 040-041. The parameter and initial condition default values for the selected crop are output by lines 139-146. If GLAI observations are input, a listing of the values is produced by the loop in lines 174-178. If the numerical solution technique is selected, values for the convergence criteria, step sizes, and AVTOL are output by lines 184-186. Also, the form of the numerical solution (value of MODEL) is output by lines 249-250. The values of XGLAI, C, B, and A at convergence of the numerical solution are output by lines 688-690. The values of three statistics (EC, NITER, and EPERC) that indicate the performance of the numerical solution technique are output by lines 691-694. EC is the total error remaining after convergence of the solution (EC will equal EB and EA upon convergence of the solutions involving parameters A and B). NITER is the total number of iterations performed before convergence was obtained for the form of the solution selected by the user. EPERC is the percentage of the total variation in the set of observed GLAI values explained by the numerical solution. Thus,

EPERC, determined in program line 687, is analogous to an R^2 value for the simulation. The simulation's estimate of grain yield is output by lines 696-697. Yield is converted (program line 695) from grams per square meter to kilograms per hectare before it is output.

The question asked by program lines 699-701 allows the user to direct the program to output a rough graph of the simulation. Daily simulated values of GLAI, AGDM, and TAGDM are plotted in the graph along with any observed values of GLAI. If the numerical solution technique was used, these daily simulated values correspond to the last iteration of the solution. The graph is produced by setting up a one-dimensional array (IPLT) that represents the range of values that a variable can assume on any given day. This range has been set at 0 to 14 for GLAI and 0 to 7000 g m^{-2} for AGDM and TAGDM. The range for each variable (GLAI, AGDM, and TAGDM) is divided into 71 parts by the dimension of the IPLT array (see program line 002). These parts can be thought of as 71 bins that the daily value for each variable can fall into. For each day following emergence of the crop, the bin in which the value of each variable (GLAI, AGDM, or TAGDM) falls is identified. Once the bin values are determined, each variable's bin values will be assigned a plotting symbol that will represent that variable. Once the plotting symbols have been assigned, the contents of IPLT are written to the output file, forming one column of the graph. When a column is formed in the output file for each consecutive day of the simulation, a complete graph of growth over the growing season will have been produced by the model.

The graph is produced in the model by lines 704-776. The symbols used to plot the variables and the borders of the graph are established in lines 704-705. Program lines 706-714 write a legend to the output file identifying the following plotting symbols:

```
SYMBOLS:
            +   SIMULATED GLAI
            *   SIMULATED DRY MASS WITH SENESCENCE
            *   SIMULATED DRY MASS WITHOUT SENESCENCE
            O   OBSERVED GLAI
            A   SIMULATED ANTHESIS DAY
            M   SIMULATED MATURITY DAY
```

Note that the same symbol is used for simulated dry mass with (AGDM) and without (TAGDM) senescence. The double function of this symbol should not cause confusion, since the greater of the two values appearing on any given day will always be TAGDM. Following the legend, a statement is written informing the user that the x-axis (the vertical axis of the output file) represents days after planting (DAP). Program lines 715-717 produce the label along the y-axis, which represents the range of GLAI values plotted in the graph.

The graph is produced by the daily loop contained in program lines 722-772. The loop starts on the first day weather data are taken and ends five days after the last day of the simulation. The end point (ILAST) of the loop is determined in line 720 from IEND, which was saved earlier in the simulation. The current day of the year (IDAY) is determined in line 723, and if IDAY is less than ISOW (line 724), no plotting will occur. If IDAY is greater than ISOW, the number of days after planting (JDAP) will be determined (line 725) and will serve as the x-axis variable for the graph. Before plotting can occur, the entire contents of the IPLT array for the current day must be blanked out

(lines 726-727), providing a blank field where the appropriate plotting symbols will be added. If IDAY equals either ISOW or ILAST, program lines 728-735 will put the starting or ending y-axis borders on the graph and will include tick marks at every tenth array element. If the current day is between ISOW and ILAST, program lines 736-737 will put the starting and ending x-axis borders, respectively, in the first and last array elements. These borders will be changed to tick marks for every tenth consecutive day after planting (lines 738-741). No additional symbols will be added if the current day is before emergence or after the last day of simulated growth (program lines 742-743). Program lines 744-749 determine which IPLT array elements contain the symbols for TAGDM, AGDM, and GLAI for the current day. The symbol for GLAI will overwrite the symbol for AGDM and TAGDM should the two symbols occur in the same IPLT array element. If the model indicates that anthesis occurs on IDAY (program line 750), the letter "A" is placed in the 69th element of the IPLT array (line 751), and the value of the indicator variable KANTH is changed from zero to one to indicate that anthesis has occurred. If simulated growth is extended beyond the maturity date (because GLAI observations were recorded beyond the maturity date), the letter "M" is placed in the 69th element of the IPLT array on the day of maturity (program lines 754-757). The absence of "M" in the graph indicates that the last day plotted is the maturity date. If one or more observed GLAI values occur on the current day, these observed values are added to the graph by program lines 759-763. After all of the appropriate symbols have been added to the IPLT array for the current day, the array contents are output to the summary by lines 765-771. In the output, the value of JDAP for every tenth day is written along the edge of the graph next to the x-axis tick marks (lines 765-766).

After the daily loop is finished, program lines 773-776 write the y-axis label involving aboveground dry mass on the graph. The graph is now complete and may be used to assess the general performance of the model. Simulated values of the state variables (SGDD, GLAI, AGDM, and TAGDM) can be output (lines 778-801) for each day of the simulation. In the output, the date is expressed not only as the current day of the year (IDAY) but also as the number of days after planting (JDAP) and the number of days after emergence (JDAE). In this listing, IDAY is corrected when the simulation passes from one year to the next (program line 798) so that the value of IDAY does not exceed the number of days in the year. The listing of daily results can be extracted from the output file and submitted to independent statistical or plotting programs for a more detailed evaluation of model performance.

Parameter Default Values. As described earlier in this document, simulations of the growth of spring wheat, corn, and sorghum can be performed using the built-in parameter default values provided in the DATA statement within program lines 057-062. If the user would like to use a completely different set of parameter default values, these can be input at execution time from a disk file associated with I/O unit 30. Input of default values is required if the user selects crop No. 4 ("A Different Crop") from the choices provided by the model (program lines 046-053). Input of default values from a disk file is optional if crop No. 1, 2, or 3 (spring wheat, corn, or sorghum, respectively) is selected. Program lines 064-071 determine if disk file input of parameter default values is optional (i.e., crop No. 1, 2, or 3 was selected by the user) or required (i.e., crop No. 4 was selected by the user). Regardless of the crop selected for simulation, the input default values will be stored in place of those originally provided for crop No. 1 (program lines 073-079). To enter values from a disk file, the user must supply the name of the disk file. GRAMI requires the contents of the disk file to be in a specific format. As shown in appendix D, the first two lines of the file can be used to identify the contents. As indicated by program lines 076-077, the first two lines of the disk file are read by the model but are not used, so they can contain any information or can be blank. Each of the remaining 20 lines in the

file must contain a parameter default value. The default values must be in the order shown in the example. The numeric format for each default is not important, since the values are read in free-format (program line 079). In each of the 20 lines, the default value must come first and must be separated from any additional information in the line by a comma or at least one blank space. The parameter descriptions shown in the example are optional—only the default values are read by the program.

After the parameter defaults have been specified, they will be listed on the terminal display by program lines 082-104. At this point, the model asks whether the user wants to change individual default values (lines 105-106). If the user answers "yes," the model will ask the user to supply the number (1-20) of the default and its new value, which can be supplied using the keyboard. After this information is entered, the list of defaults will be redisplayed to verify that the change has been made. Values can be changed by this procedure as many times as the user desires. When the changes are complete, the default values will be loaded for their respective variables (program lines 119-138).

Input of GLAI Values. The remaining I/O activity involves the input of observed GLAI values (program lines 149-171). The model will ask whether the user wants to input the values from the keyboard or a disk file (program lines 153-154). To answer the question, the user must supply a value (either 1 or 2) for the indicator variable KENTER. If the user specifies KENTER = 1, the GLAI observations will be read from a disk file (program lines 157-162). The program will then ask the user to input the name of the disk file associated with I/O unit 40, from which the GLAI values will be read. As the values are read, the program counts them to determine the total number of GLAI observations (NOBS). GRAMI requires the values in the disk file to be in a specific format and in chronological order according to observation date. Each line of the file should contain the day of the year of the observation (IDATE) and the observed GLAI value, separated by a comma or at least one blank space. The value for IDATE must be an integer. The GLAI value (OBLAI) does not have to contain a decimal point if it is a whole number. The utility program CONV_OB (in appendix C) can be used to convert GLAI observations in existing data sets into the required format and to arrange them chronologically. Like CONV_WX, CONV_OB is self-explanatory when executed and, therefore, will not be described further.

If the user specifies KENTER = 2, the GLAI values can be input from the keyboard (program lines 164-170). Inputting from the keyboard is usually convenient when there is only a small number of observations. First, the user must supply the value for NOBS. Once this is done, the program will ask for each GLAI value and the date each value was observed. The observed GLAI values must be entered in chronological order.

Examples for Various Crops

In this section, the results of running GRAMI for four crops are presented to demonstrate the performance of the model. The first three examples involve basic applications of the model utilizing the built-in parameter default values for spring wheat, corn, and grain sorghum. The fourth example involves the use of disk file input of parameter defaults for winter wheat, a crop that does not have parameter defaults built into the model. These examples should provide additional information that may assist the user's application of the model.

Values of GLAI determined through destructive sampling are used in place of remotely sensed estimates of GLAI to drive the numerical solutions in these examples. The destructive sampling method was chosen to prevent errors from being introduced into the simulations as a result of inaccuracies in estimating GLAI from remotely sensed data. Such inaccuracies would confound the assessment of the model's ability to simulate crop growth and yield.

Spring Wheat

The data for this crop were collected in 1983 at Phoenix, Arizona (Jackson and Pinter 1986). These data were reported by Maas et al. (1989) in a modeling effort that involved multiple wheat varieties and irrigation treatments; however, only the control treatment (no water stress) data for one spring wheat variety (Yecora 70) were used here.

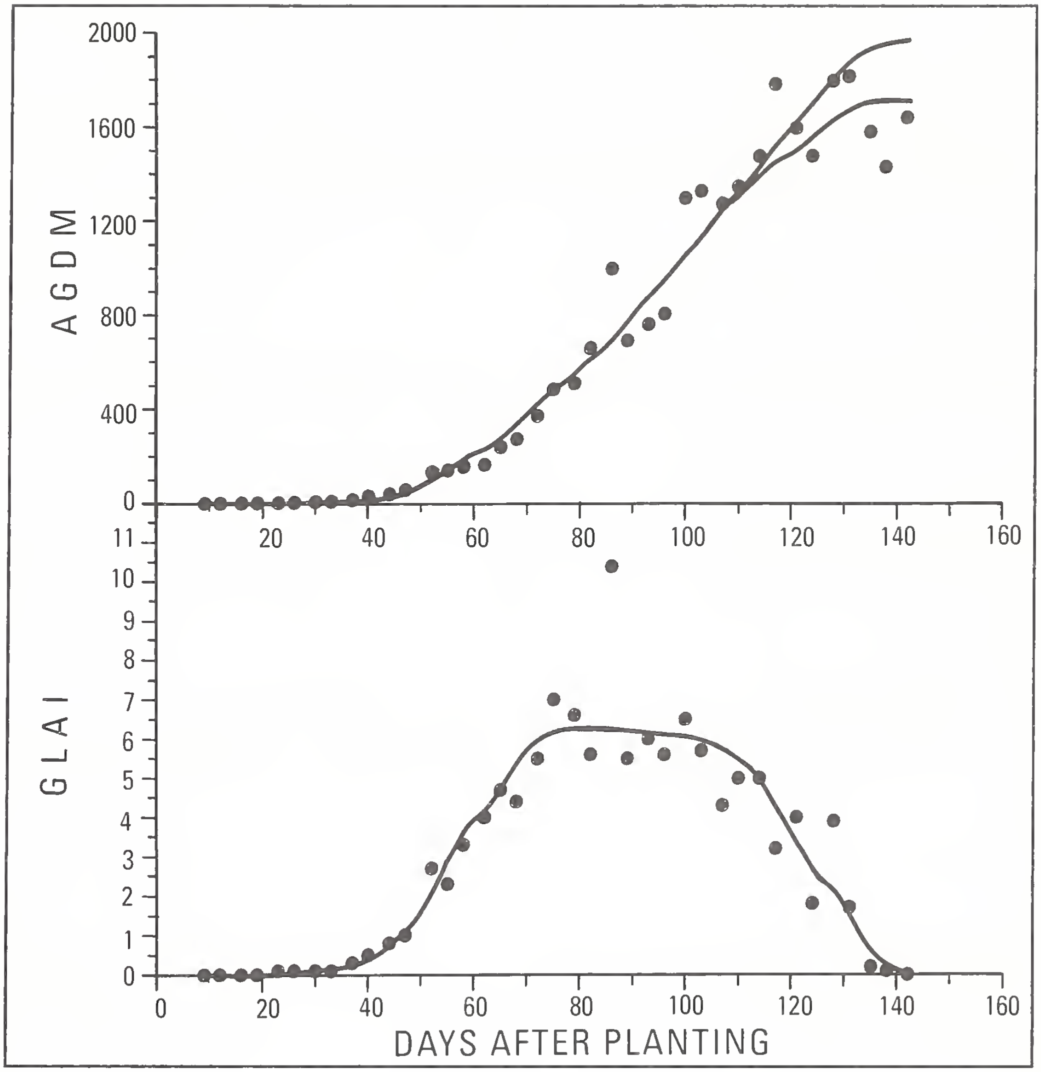
Seeds were planted into dry soil in late December, and a late irrigation was used to initiate growth. For the sake of simplicity, the planting date used in the model simulation was assumed to be 1 January 1983. Weather observations were collected at the experiment site, and the program CONV_WX was used to put the temperature and irradiance data into the proper format before inputting. GLAI was determined twice weekly using destructive sampling. The program CONV_OB was used to format the observed GLAI data.

The parameter default values established for this simulation are contained in lines 057-058 of the program listing. The default values for EX, EF, SLA, YF, and BASET were selected as being representative for this crop and therefore were not changed. The default values for SDDEM, SDDAN, and SDDPM were estimated from the daily temperature data and observed dates of crop emergence, anthesis, and maturity. The default values for XGLAI, C, B, and A were determined using equation 1.

When the numerical solution technique was selected, the model responded that model form 3 (involving XGLAI, C, and B) should be appropriate for this simulation. However, since GLAI observations were taken many times throughout the growing season, model form 4 (the full numerical solution) was selected. The simulation required 740 iterations of the numerical solution to converge. The values of XGLAI, C, B, and A that resulted in convergence were 0.0478798, 65.2, 0.003650, and 0.039952, respectively. These values are relatively close to the respective default values (0.01, 70.0, 0.003, and 0.04) used to start the numerical solution.

Results of the simulation are presented in figure 8. The simulation explained 84 percent of the total variation in observed GLAI. The unexplained variation is largely due to measurement error. The simulation of AGDM (lower dry mass curve) accounts for leaf senescence and passes through the biomass observations in a reasonable manner over the duration of the growing season. The simulation of total AGDM (upper dry mass curve) does not account for leaf senescence and therefore overestimates the observed biomass near the end of the growing season.

Figure 8
Comparisons of simulated GLAI and AGDM (solid lines) to observations (circles) for spring wheat grown at Phoenix, AZ. Lower AGDM curve accounts for leaf senescence, while the upper AGDM curve does not.



In GRAMI the AGDM simulation is not numerically fit to biomass observations (AGDM observations are not a model input). However, when the GLAI simulation reasonably fits the GLAI observations, the associated AGDM simulation reasonably approximates the biomass observations. This correspondence between the GLAI and AGDM simulations implies that the crucial activity in the model is the accurate simulation of leaf area.

The grain yield estimated by the simulation was 6747 kg ha $^{-1}$. The observed yield reported for the experimental field was 7120 kg ha $^{-1}$ Although it is not possible to statistically determine model accuracy from a single observation, the simulated yield was only about 5 percent lower than the observed yield. In a large-area yield-estimation program involving many fields, an unbiased error of less than 10 percent for individual fields should be quite acceptable.

Corn (Maize)

The data for this crop were collected in 1985 at Weslaco, TX. These data were used in earlier modeling efforts reported by Maas et al. (1985, 1989) involving irrigated and dryland treatments. However, only the control treatment (no water stress) data were used here.

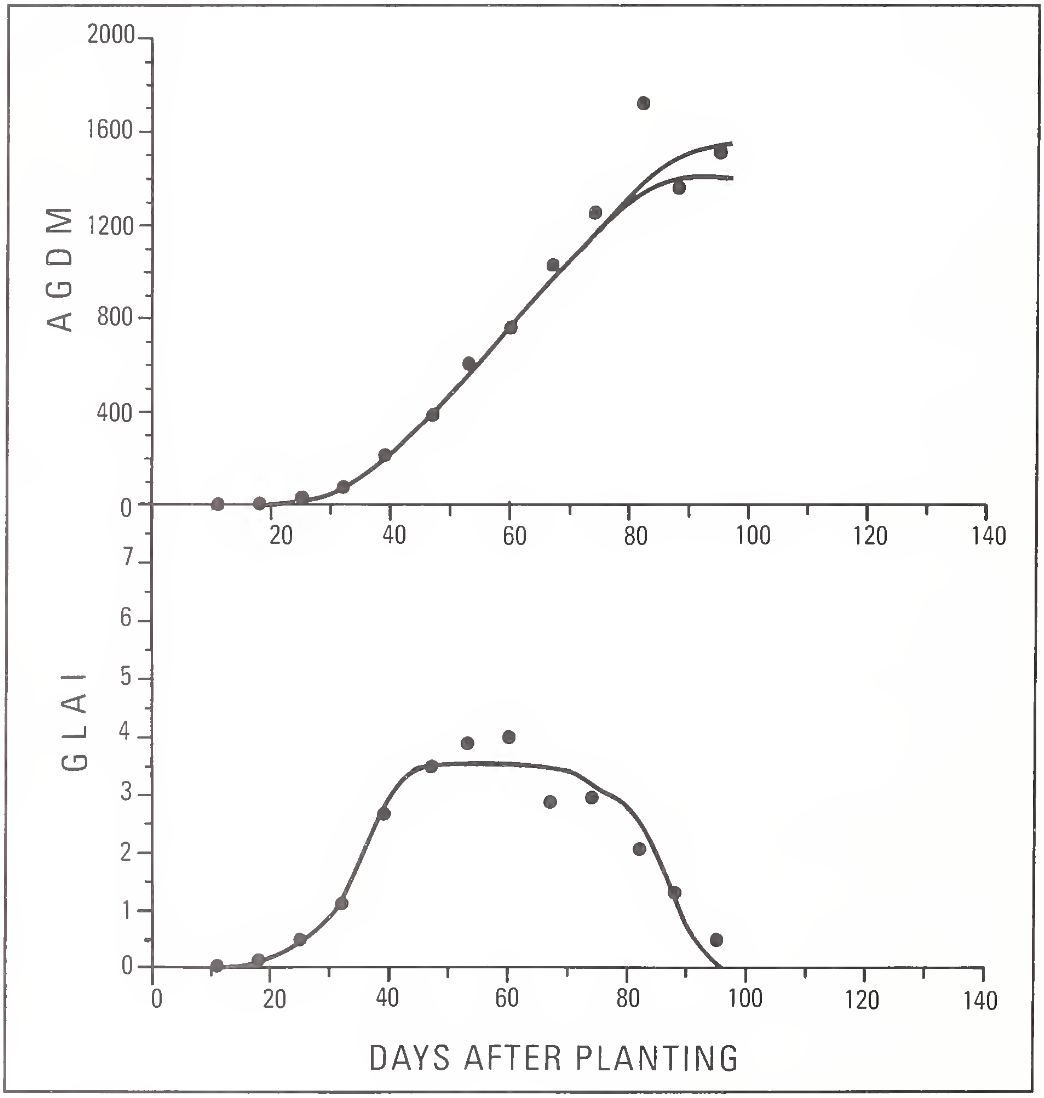
Seeds of white corn (cv. Thermal) were planted on 21 May 1985 (day 143). Weather observations were made at the site starting on day 140. GLAI was determined weekly using destructive sampling. As in the previous example, the programs CONV_WX and CONV_OB were used to format the weather data and GLAI input data, respectively.

The parameter default values established for this simulation are contained in lines 059-060 of the program listing. The default values for EX, EF, SLA, YF, and BASET were selected as being representative for this crop. The value for YF (0.5) is less than that typical for the more widely grown yellow field corns (YF ≅ 0.75) and reflects the relatively smaller yield potential of white corn. The default values for SDDEM, SDDAN, SDDPM, XGLAI, C, B, and A were estimated in the same manner as in the previous example.

The full numerical model (involving XGLAI, C, B, and A) was selected for this simulation. A total of 665 iterations were required for the solution to converge. The values of XGLAI, C, B, and A that resulted in convergence were 0.0078283, 50.4, 0.004146, and 0.049941, respectively. These values are relatively close to the respective default values (0.01, 50.0, 0.003, and 0.05) used to start the numerical solution.

Results of the simulation are presented in figure 9. The simulation explained 88 percent of the total variation in observed GLAI. The simulations of AGDM (lower dry mass

Figure 9
Comparisons of simulated GLAI and AGDM (solid lines) to observations (circles) for corn (maize) grown at Weslaco, TX. Lower AGDM curve accounts for leaf senescence, while the upper AGDM curve does not.



curve) and TAGDM (upper dry mass curve) pass through the biomass observations in a reasonable manner, with the effects of senescence appearing relatively late in the growing season. The grain yield of corn estimated by the simulation (4301 kg ha^{-1}) was only about 2 percent higher than the observed yield (4215 kg ha^{-1}).

Grain Sorghum

The data for this crop were collected in 1987 from a study conducted at Weslaco, TX. The study involved the sorghum variety ORO G-XTRA planted in a replicated, randomized design with two tillage treatments (conventional and no-till) and four fertilization rates (0, 40, 80, and 120 lb N $acre^{-1}$). However, only the control treatment (conventional tillage, no additional fertilization applied) data were used here.

Planting occurred on 20 March 1987 (day 79). Weather data were collected at the field location starting on day 60. Using destructive sampling methods, GLAI was determined weekly until anthesis, and biweekly thereafter. The utility programs CONV_WX and CONV_OB were used to format the weather data and GLAI data, respectively.

The parameter default values established for this simulation are contained in lines 061-062 of the program listing. The default values for EX, EF, SLA, YF, and BASET were selected as being representative for this crop. The default values for SDDEM, SDDAN, SDDPM, XGLAI, C, B, and A were determined in the same manner as in the two previous examples.

Model form 4 (involving XGLAI, C, B, and A) was selected for the simulation in this example. A total of 900 iterations were required for convergence of the numerical solution. The values of XGLAI, C, B, and A that resulted in convergence were 0.00732973, 58.1, 0.003344, and 0.067939, respectively. These convergence values are relatively close to the respective default values (0.01, 50.0, 0.003, and 0.067) used to start the numerical solution.

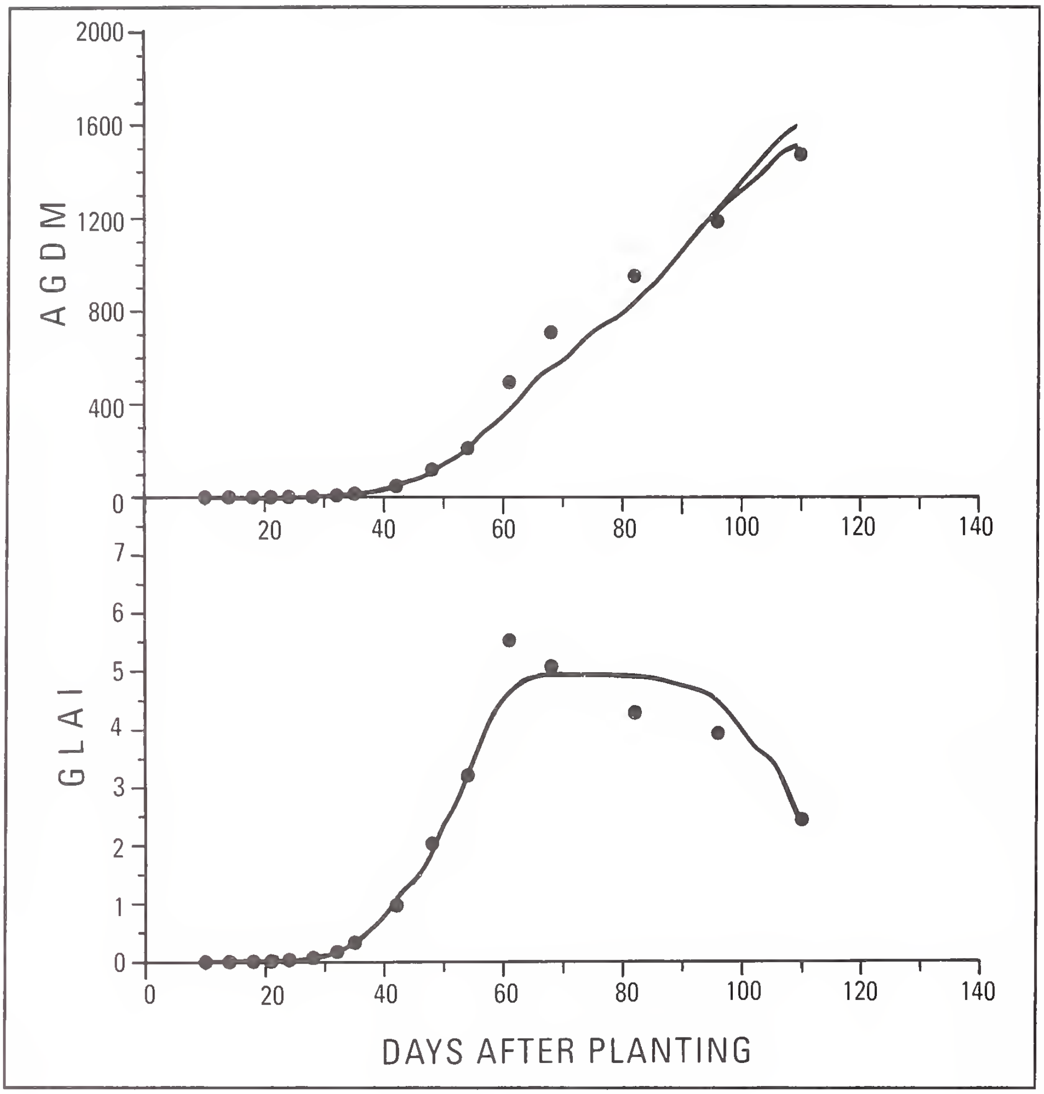
Results of the simulation are presented in figure 10. The simulation explained 91 percent of the total variation in observed GLAI. The simulations of AGDM (lower dry mass curve) and TAGDM (upper dry mass curve) pass through the biomass observations in a reasonable manner. The effects of senescence are apparent only near maturity of the crop because grain sorghum characteristically maintains a large portion of its total leaf area over the duration of the grain-filling period. The grain yield of grain sorghum estimated by the simulation (7005 kg ha^{-1}) was only about 4 percent lower than the observed yield (7316 kg ha^{-1}).

As stated earlier, the model's ability to estimate yield cannot be statistically determined from a single comparison of simulated and observed values. However, the model's reasonable yield estimations for spring wheat, corn, and grain sorghum tend to indicate that, when the GLAI and AGDM simulations reasonably match what occurs in the field, the yield estimate will also be reasonably accurate.

Winter Wheat

Winter wheat is planted in the fall and typically begins growth prior to the onset of winter. The cold temperatures of winter halt the growth of the crop and often kill a portion of the aboveground biomass. Where winters are severe, all of the aboveground parts of the plant can be killed. Even the subsurface crowns of plants may be killed, reducing the effective plant population in the field. Where winters are milder, a large portion of the aboveground biomass may survive. The quantification of the winterkill effect in crop

Figure 10
Comparisons of simulated GLAI and AGDM (solid lines) to observations (circles) for grain sorghum grown at Weslaco, TX. Lower AGDM curve accounts for leaf senescence, while the upper AGDM curve does not.



growth simulation models is difficult, since winterkill typically occurs irregularly in a field (Fowler 1979) and is not easily related to commonly available environmental data.

Once temperatures increase in the spring, growth of the surviving winter wheat crop resumes. In areas where all of the aboveground biomass has been killed, the new growth resembles that of a newly emerged spring wheat crop. Even when a portion of the crop canopy survives the winter, the leaf and stem tissue present between maximum GLAI and maturity is primarily produced in the spring. Thus, it may be appropriate to ignore the fall growth and the winter survival by modeling only the spring growth of a winter wheat crop.

The data for demonstrating the modeling of spring growth were collected as part of the 1978-79 Tri-State Winter Wheat Study, which involved 10 fields in 4 States in the U.S. Great Plains. These data were used in an earlier model validation study reported by Maas and Arkin (1980). This example will involve the data obtained for the field located at Temple, TX.

Seeds of the cv. Sturdy were planted on 25 October 1978 (day 298). Following emergence on day 303, the crop grew vegetatively until approximately 1 January, when cooler temperatures led to crop dormancy. The winter was short and mild; only a small fraction of the aboveground biomass was lost to frosts. Growth of the crop resumed in late January. Frequent rains allowed the crop to grow unstressed during the experiment. Weather data were collected at a location near the field, starting before the date of planting. GLAI was determined using destructive sampling on an irregular schedule; most observations occurred during the period of spring growth.

To simulate spring growth, GRAMI requires the user to select a day during winter dormancy as the planting date. For simplicity, 1 January 1979 was used in this example, although any day during the dormancy could have been chosen. Because the model does not have default parameter values built in for winter wheat, the default values for this simulation were input from a disk file (values shown in appendix D). The value for SDDEM was arbitrarily selected as 100 GDDs. In this simulation, the emergence date should approximately correspond to the date when spring growth resumes. The values for SDDAN and SDDPM were estimated as the sum of the GDDs starting from 1 January to the observed dates of anthesis and maturity. The default values for the parameters EX, EF, SLA, YF, and BASET were the same as those used earlier in simulating spring wheat. The default values for XGLAI, C, B, and A were determined using equations 1a-d.

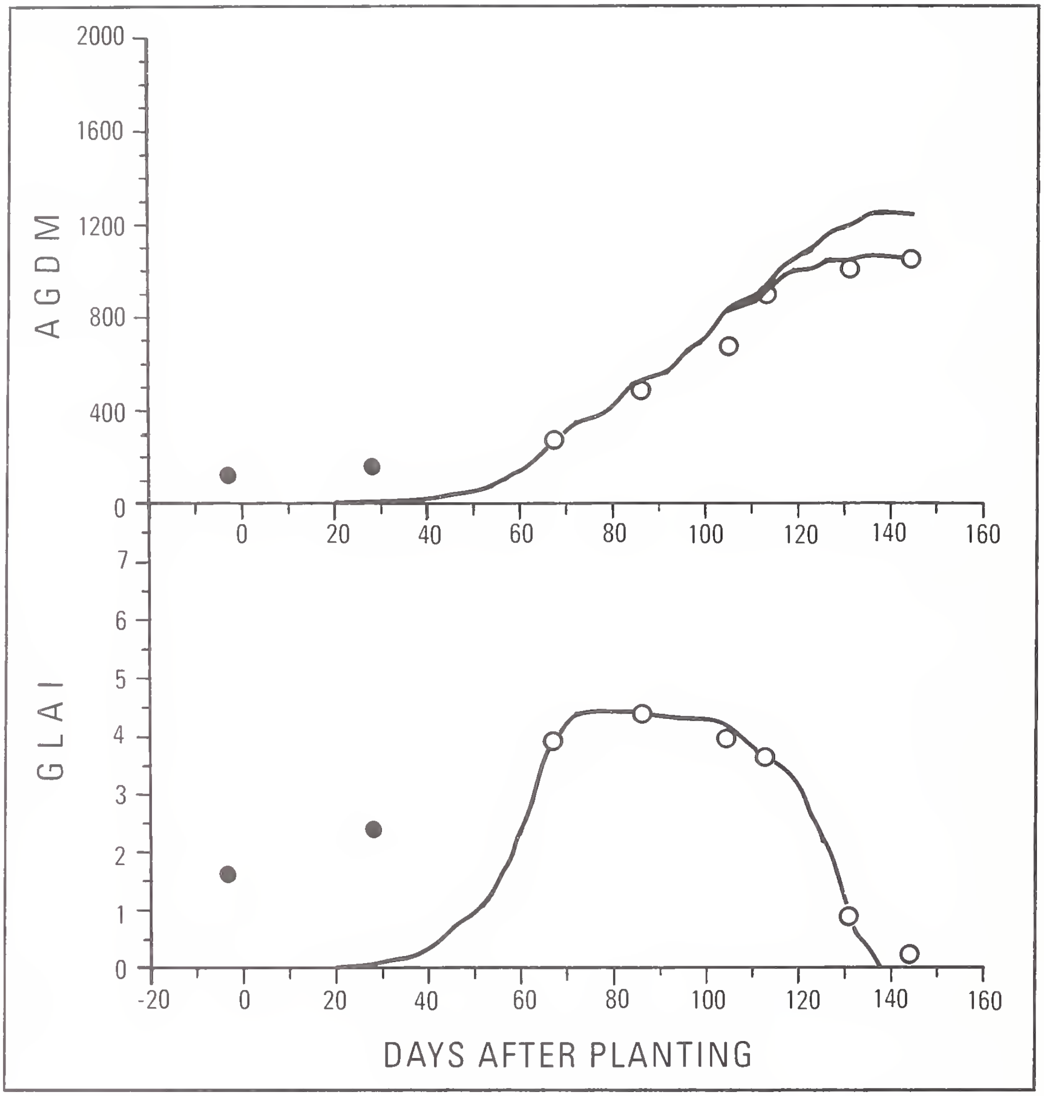
GLAI observations used in this simulation were obtained through destructive sampling. Had all the fall growth been killed during the winter, all of the GLAI observations could have been used. However, when a portion of the fall growth survives the dormancy period (as happened in this example), the surviving biomass can affect the GLAI values obtained soon after growth resumes in the spring. Once maximum GLAI is attained, the aboveground biomass is dominated by spring growth, and the observations obtained during or after this time are minimally affected by the surviving biomass. Therefore, only those observations from the time of maximum GLAI or later were used in the simulation.

When the model was executed, 1 January was input as the planting date. Crop No. 4 ("A Different Crop") was selected from the menu of choices, and the parameter default values were then input using a disk file. Model form 4 (involving XGLAI, C, B, and A) was selected for the numerical solution. The simulation required 584 iterations for the numerical solution to converge. The values of XGLAI, C, B, and A that resulted in convergence were 0.1338987, 65.6, 0.004594, and 0.088472, respectively.

Results of the simulation are presented in figure 11. The simulation explained 97.6 percent of the total variation in the GLAI observations. The simulation of AGDM (including the effects of senescence) passes through the biomass observations dominated by spring growth (open circles) in a reasonable manner. As might be expected, the TAGDM simulation tends to overestimate the observed biomass near the end of the growing season.

The grain yield estimated by the simulation (3602 kg ha^{-1}) was only about 5 percent lower than the observed yield (3807 kg ha^{-1}). The results of this simulation indicate that modeling the spring growth of winter wheat may be an appropriate approach for estimating the yield of this crop.

Figure 11
Comparisons of simulated GLAI and AGDM (solid lines) to observations (circles) for winter wheat grown at Temple, TX. Open circles denote values dominated by spring growth; filled circles denote values dominated by fall growth. Only GLAI observations denoted by open circles were used in the numerical solution. Lower AGDM curve accounts for leaf senescence, while the upper AGDM curve does not.



Summary

The methodology described in this document represents a novel combination of crop modeling and remote sensing such that the strengths of one technology make up for the weaknesses of the other. The methodology takes advantage of the dependence of model performance on parameters and initial conditions that can be evaluated using GLAI observations and numerical analysis procedures. Through reinitialization and reparameterization, the model's simulation of GLAI can be brought into agreement with observed GLAI. As indicated by the examples for various crops, aboveground dry mass and grain yield can be reasonably estimated when the GLAI simulation is fit to a set of GLAI observations by the numerical solution technique.

The version of GRAMI described in this document is relatively unsophisticated in simulating the growth of crop plants. Because the model's data requirements are rather modest and the model is easy to execute, it is hoped that this document may inspire other researchers to further develop this methodology.

References

Arkin, G.F., S.J. Maas, and C. Richardson. 1980. Forecasting grain sorghum yields using simulated weather data and updating techniques. Transactions of ASAE 23:676-680.

Arkin, G.F., W.D. Rosenthal, and W.R. Jordan. 1983. A sorghum leaf area model. ASAE Technical Paper No. 83-2098, 25 pp.

Arkin, G.F., R.L. Vanderlip, and J.T. Ritchie. 1976. A dynamic grain sorghum growth model. Transactions of ASAE 19:622-630.

Arkin, G.F., C.L. Wiegand, and H. Huddleston. 1977. The future role of a crop model in large area yield estimation. *In* Proceedings of the Crop Modeling Workshop, pp. 87-116. USDC-NOAA-EDIS-CEAS, Columbia, MO.

Asrar, G., E.T. Kanemasu, R.D. Jackson, and P.J. Pinter, Jr. 1985. Estimation of total above-ground phytomass production using remotely sensed data. Remote Sensing of Environment 17:211-220.

Brakke, T.W., and E.T. Kanemasu. 1979. Estimated winter wheat yields from Landsat MSS using spectral techniques. *In* Proceedings of the 13th International Symposium on Remote Sensing of Environment, pp. 629-641. ERIM, Ann Arbor, MI.

Charles-Edwards, D.A. 1982. Physiological determinants of crop growth. 161 pp. Academic Press, New York.

Charles-Edwards, D.A., D. Doley, and G.M. Rimmington. 1986. Modelling plant growth and development. 235 pp. Academic Press, Orlando, FL.

Daughtry, C.S.T., K.P. Gallo, and M.E. Bauer. 1983. Spectral estimates of solar radiation intercepted by corn canopies. Agronomy Journal 75:527-531.

Delecolle, R., and M. Guerif. 1988. Introducing spectral data into a plant process model for improving its prediction ability. *In* Proceedings of the 14th International Colloquium on Spectral Signatures of Objects in Remote Sensing, pp. 125-127. ESA, Aussois, France.

Fowler, D.B. 1979. Selection for winterhardiness in wheat: II. Variation within field trials. Crop Science 19:773-775.

Friend, D.J.C. 1966. The effects of light and temperature on the growth of cereals. *In* F.L. Milthorpe and J.D. Ivins, eds., The Growth of Cereals and Grasses, pp. 181-199. Butterworths, London, England.

Garrity, D.P., C.Y. Sullivan, and D.G. Watts. 1984. Changes in grain sorghum stomatal and photosynthetic response to moisture stress across growth stages. Agronomy Journal 24:441-446.

Gibson, P.T., and K.F. Schertz. 1977. Growth analysis of a sorghum hybrid and its parents. Crop Science 17:387-391.

Hanway, J.J., and W.A. Russell. 1969. Dry-matter accumulation in corn (*Zea mays* L.) plants: Comparisons among single-cross hybrids. Agronomy Journal 61:947-951.

Hatfield, J.L., E.T. Kanemasu, G. Asrar, and others. 1985. Leaf-area estimates from spectral measurements over various planting dates of wheat. International Journal of Remote Sensing 6:167-175.

Hodges, T., and E.T. Kanemasu. 1977. Modeling daily dry matter production of winter wheat. Agronomy Journal 69:974-978.

Jackson, R.D. 1983. Spectral indices in *n*-space. Remote Sensing of Environment 13:409-421.

Jackson, R.D., and P.J. Pinter, Jr. 1986. Spectral response of architecturally different wheat canopies. Remote Sensing of Environment 20:43-56.

Jackson, R.D., P.J. Pinter, Jr., R.J. Reginato, and S.B. Idso. 1980. Hand-held radiometry. U.S. Department of Agriculture Agricultural Reviews and Manuals ARM-W-19, 66 pp.

Jewiss, O.R. 1966. Morphological and physiological aspects of growth of grasses during the vegetative phase. *In* F.L. Milthorpe and J.D. Ivins, eds., The Growth of Cereals and Grasses, pp. 39-53. Butterworths, London, England.

Kanemasu, E.T., G. Asrar, and M. Fuchs. 1985. Application of remotely sensed data in wheat growth modelling. *In* W. Day and R.K. Atkin, eds., Wheat Growth Modelling, pp. 357-369. Plenum Press, New York.

Kanemasu, E.T., A. Feyerherm, B. Latchford, and others. 1982. Crop model development using remotely sensed thermal and optical data. 227 pp. Development Planning and Research Associates, Manhattan, KS.

Kauth, R.J., and G.S. Thomas. 1976. The tasseled cap—A graphic description of the spectral-temporal development of agricultural crops as seen by Landsat. *In* Proceedings of the Symposium on Machine Processing of Remotely Sensed Data, pp. 41-51. LARS, West Lafayette, IN.

Maas, S.J. 1987. Growth model for gramineous crops utilizing remotely-sensed data. *In* Agronomy Abstracts, p. 15. ASA, Madison, WI.

Maas, S.J. 1988a. Use of remotely-sensed information in agricultural crop growth models. Ecological Modelling 41:247-268.

Maas, S.J. 1988b. Using satellite data to improve model estimates of crop yield. Agronomy Journal 80:655-662.

Maas, S.J. 1988c. Using field observations to improve growth model performance. *In* Abstracts of the Workshop on Crop Simulation, p. 31. Biological Systems Simulation Group, Gainesville, FL.

Maas, S.J., and G.F. Arkin. 1978. User's guide to SORGF: A dynamic grain sorghum growth model with feedback capacity. TAES Program and Model Documentation No. 78-1, 107 pp.

Maas, S.J., and G.F. Arkin. 1980. Initial validation of a winter wheat simulation model. ASAE Technical Paper No. 80-4010, 7 pp.

Maas, S.J., R.D. Jackson, S.B. Idso, and others. 1988. Crop simulation model guided by remotely-sensed LAI and CWSI data. *In* Agronomy Abstracts, pp. 22-23. ASA, Madison, WI.

Maas, S.J., R.D. Jackson, S.B. Idso, and others. 1989. Incorporation of remotely-sensed indicators of water stress in a crop growth simulation model. *In* Proceedings of the 19th Conference on Agricultural and Forest Meteorology, pp. 228-231. AMS, Charleston, SC.

Maas, S.J., A.J. Richardson, C.L. Wiegand, and P.R. Nixon. 1985. Use of plant, spectral and weather data in modeling corn growth. *In* Proceedings of the 19th International Symposium on Remote Sensing of Environment, pp. 167-186. ERIM, Ann Arbor, MI.

Matthews, R.V. 1985. An overview of the 1985 corn, cotton, soybean, and wheat objective yield surveys. 25 pp. U.S. Department of Agriculture, Statistical Reporting Service, Statistical Research Division, Washington, DC.

McCree, K.J., C.E. Kallsen, and S.G. Richardson. 1984. Carbon balance of sorghum plants during osmotic adjustment to water stress. Plant Physiology 76:898-902.

Pollock, R.B., and E.T. Kanemasu. 1979. Estimating leaf area index of wheat with Landsat data. Remote Sensing of Environment 8:307-312.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. 1986. Numerical recipes: The art of scientific computing. 818 pp. Cambridge University Press, Cambridge, England.

Redelfs, M.S., L.R. Stone, E.T. Kanemasu, and M.B. Kirkham. 1987. Greenness-leaf area index relationships of seven row crops. Agronomy Journal 79:254-259.

Richardson, A.J., and C.L. Wiegand. 1977. Distinguishing vegetation from soil background information. Photogrammetric Engineering and Remote Sensing 43:1541-1552.

Richardson, A.J., C.L. Wiegand, G.F. Arkin, and others. 1982. Remotely-sensed spectral indicators of sorghum development and their use in growth modeling. Agricultural Meteorology 26:11-23.

Strand, B.W. 1981. Spatial scale of crop-yield models. USDA ESS Staff Report No. AGESS810320, 13 pp.

Vanderlip, R.L., and G.F. Arkin. 1977. Simulating accumulation and distribution of dry matter in grain sorghum. Agronomy Journal 69:917-923.

Wanjura, D.F., and J.L. Hatfield. 1985. Spectral procedures for estimating crop biomass. Transactions of ASAE 28:922-927.

Wiegand, C.L., and J.L. Hatfield. 1988. The spectral-agronomic multisite-multicrop analyses (SAMMA) project. International Archives of Photogrammetry and Remote Sensing 27:696-706.

Wiegand, C.L., A.J. Richardson, and E.T. Kanemasu. 1979. Leaf area index estimates for wheat from Landsat and their implications for evapotranspiration and crop modeling. Agronomy Journal 71:336-342.

Wilson, D.R., C.H.M. van Bavel, and K.J. McCree. 1980. Carbon balance of water-deficit grain sorghum plants. Crop Science 20:153-159.

Appendix A: Glossary of Variables

Name	Type	Definition
A	REAL	Parameter in the equation for stem partitioning fraction (SPF)
AE1	REAL	Total errors associated with the portion of the solution involving parameter A for the current (4th) and three previous (3d, 2d, and 1st, with the 3d being the most recent) iterations
AE2	REAL	
AE3	REAL	
AE4	REAL	
AGDM	REAL	Aboveground dry mass on the current day (in g m^{-2})
AMAX	REAL	Maximum value allowed for A in the numerical solution
AMIN	REAL	Minimum value allowed for A in the numerical solution
ANS	CHAR	Variable that stores the user's response to a yes/no question asked by the program
AP1	REAL	Values of A for the current (4th) and three previous (3d, 2d, and 1st, with the 3rd being the most recent) iterations of the portion of the numerical solution involving parameter A
AP2	REAL	
AP3	REAL	
AP4	REAL	
APAR	REAL	Absorbed photosynthetically active radiation on the current day (in MJ m^{-2})
ASTB	REAL	Value of B stored at the start of the current iteration involving the parameter A
ASTC	REAL	Value of C stored at the start of the current iteration involving the parameter A
ASTG	REAL	Value of XGLAI stored at the start of the current iteration involving the parameter A
AVTOL	REAL	Average tolerable error established by the user
B	REAL	Parameter in the equation for stem partitioning factor (SPF)
BASET	REAL	Base temperature for computing degree-days in deg C
BE1	REAL	Total errors associated with the portion of the numerical solution involving the parameter B for the current (4th) and three previous (3d, 2d, and 1st) iterations
BE2	REAL	
BE3	REAL	
BE4	REAL	
BMAX	REAL	Maximum value allowed for B in the numerical solution

BMIN	REAL	Minimum value allowed for B in the numerical solution
BP1 BP2 BP3 BP4	REAL REAL REAL REAL	Values of B for the current (4th) and three previous (3d, 2d, and 1st, with the 3rd being the most recent) iterations of the portion of the numerical solution involving parameter B
BSTC	REAL	Value of C stored at the start of the current iteration involving the parameter B
BSTG	REAL	Value of XGLAI stored at the start of the current iteration involving the parameter B
C	REAL	Expected leaf lifespan (in days)
CE1 CE2 CE3 CE4	REAL REAL REAL REAL	Total errors associated with the portion of the numerical solution involving parameter C for the current (4th) and three previous (3d, 2d, and 1st) iterations
CMAX	REAL	Maximum value allowed for C in the numerical solution
CMIN	REAL	Minimum value allowed for C in the numerical solution
CONVA	REAL	Convergence criterion for the portion of the numerical solution involving parameter A
CONVB	REAL	Convergence criterion for the portion of the numerical solution involving parameter B
CONVC	REAL	Convergence criterion for the portion of the numerical solution involving parameter C
CONVG	REAL	Convergence criterion for the portion of the numerical solution involving XGLAI
CP1 CP2 CP3 CP4	REAL REAL REAL REAL	Values of C for the current (4th) and three previous (3d, 2d, and 1st, with the 3d being the most recent) iterations of the portion of the numerical solution involving parameter C
CSTG	REAL	Value of XGLAI stored at the start of the current iteration involving parameter C
DAGDM	REAL	Increase in total aboveground dry mass on the current day (in g m^{-2})
DELA	REAL	Bracketing search step size involving parameter A
DELB	REAL	Bracketing search step size involving parameter B
DELC	REAL	Bracketing search step size involving parameter C

DGLAI	REAL	Increase or decrease in GLAI on the current day
DIST	REAL	Value representing the maximum separation (in GDDs) between GLAI observations; used to determine which form of the numerical solution should be used
E	REAL	Total error associated with the current iteration
E1 E2 E3 E4	REAL REAL REAL REAL	Total errors in subroutines PARAB, REVERS, and RESEL3 for the current (4th) and three previous (3d, 2d, and 1st, with the 3d being the most recent) iterations
EA	REAL	Total error associated with the current iteration of the portion of the solution involving parameter A
EB	REAL	Total error associated with the current iteration of the portion of the solution involving parameter B
EC	REAL	Total error associated with the current iteration of the portion of the solution involving parameter C
EF	REAL	Parameter that relates the daily increase in biomass to the daily absorption of PAR (in g MJ^{-1})
EMAX	REAL	Sum of the observed GLAI values
ENEG	REAL	Absolute value of the sum of the negative differences between observed and simulated GLAI values
EPERC	REAL	Percentage of the total variation in observed GLAI explained by the simulation
EPOS	REAL	Sum of the positive differences when simulated GLAI is subtracted from observed GLAI
ERROR	REAL	Difference between the observed and simulated value of GLAI on a given day
EX	REAL	Extinction coefficient in the equation relating absorbed PAR to leaf area index
GLAI	REAL	Green (living) leaf area index
HEDLIN	CHAR	Array containing the contents written as the headline on the simulation summary
I	INTG	Counter (array index) used in program loops
IAGDM	INTG	Variable containing the plotting symbol for AGDM
IANTH	INTG	Variable containing the plotting symbol for anthesis

IBLNK	INTG	Variable containing the plotting symbol for a blank space
ICAT	INTG	Variable containing the plotting symbol for a vertical line segment
IDATE	INTG	Day of year on which a GLAI observation occurred
IDAY	INTG	Current day of year in the simulation
IDEF	INTG	Number (1-20) of the parameter default being changed
IEMER	INTG	Day of year when crop emergence occurred
IEND	INTG	Last day in the crop simulation
IGLAI	INTG	Variable containing the plotting symbol for GLAI
II	INTG	Counter used in program loops
ILAST	INTG	Five days after the last day in the crop simulation
ILINE	INTG	Variable containing the plotting symbol for a horizontal line segment
IPLT	CHAR	Element in the one-dimensional array used to produce a rough graph of the simulation results
IPMAT	INTG	Variable containing the plotting symbol for maturity
IOBS	INTG	Variable containing the plotting symbol for observed GLAI
ISOW	INTG	Day of year when planting occurred
ISTRT	INTG	Day of year when the weather data starts
ITA ITB ITC ITG	INTG INTG INTG INTG	Number of the current iteration in the portion of the numerical solution involving parameter A, B, or C or the initial condition XGLAI
IX	INTG	Counter (array index) in a computational loop
IYEAR	INTG	Year when planting occurred
J	INTG	Counter (array index) in a computational loop
JCROP	INTG	Index indicating the crop selected for simulation
JDAE	INTG	Number of days after crop emergence
JDAP	INTG	Number of days after planting

JMX	INTG	Estimate of when maximum GLAI occurred; used in suggesting which form of the numerical solution to use
JPM	INTG	Estimate of the crop maturity date; used in suggesting which form of the numerical solution to use
KANTH	INTG	Variable that indicates whether anthesis has occurred
KAPAB	INTG	Variable that indicates whether parabolic interpolation should be used for parameter A
KBPAB	INTG	Same as KAPAB, but for parameter B
KCPAB	INTG	Same as KAPAB, but for parameter C
KEMER	INTG	Variable that indicates whether crop has emerged
KENTER	INTG	Variable holding the response indicating whether data are to be input from a diskfile or the keyboard
KPMAT	INTG	Variable that indicates whether maturity has occurred
MAXTRY	INTG	Maximum number of iterations allowed for any portion of the numerical solution
MNDAY	INTG	Date of the earliest GLAI observation
MODEL	INTG	Form (0-4) of the numerical solution to be used
MXDAY	INTG	Date of the latest GLAI observation
NDAYS	INTG	Number of days of weather data
NITER	INTG	Total number of iterations used in the solution
NO1	CHAR	Variable containing the uppercase symbol for "no"
NO2	CHAR	Variable containing the lowercase symbol for "no"
NOBS	INTG	Number of GLAI observations
NOBSX	INTG	Estimated number of GLAI observations between maximum GLAI and maturity
NYEAR	INTG	Number of days in the year of planting
OBLAI	REAL	Observed value of GLAI
OLDE	REAL	Error value saved from the previous iteration
OLDG	REAL	XGLAI value saved from the previous iteration
P	REAL	Current value of parameter A, B, or C in subroutines REPLAC or REVERS

P1 P2 P3 P4	REAL REAL REAL REAL	Current (4th) and and three previous (3d, 2d, and 1st, with the 3d being the most recent) values of parameter A, B, or C in subroutines REPLAC, REVERS, PARAB, and RESEL3
PAR	REAL	Photosynthetically active radiation (in MJ m^{-2} day^{-1})
PARAB	SUBR	Subroutine that calculates the midpoint of the parabola passing through three points
REPLAC	SUBR	Subroutine that includes the newest point and removes the oldest point in the bracketing search
RESEL3	SUBR	Subroutine that selects three of four points to make up the new parabola
REVERS	SUBR	Subroutine that switches the order of points as a result of a reversal in the search direction
SDD	REAL	Growing degree-days for the current day (in deg C)
SDDAN	REAL	Sum of GDDs from planting to anthesis (in deg C)
SDDEM	REAL	Sum of GDDs from planting to emergence (in deg C)
SDDPM	REAL	Sum of GDDs from planting to maturity (in deg C)
SGDD	REAL	Sum of GDDs, planting to the current day (in deg C)
SLA	REAL	Specific leaf area of canopy leaf tissue (in m^2 g^{-1})
SLOPE	REAL	Local slope of the error function involving XGLAI
SPF	REAL	Stem partitioning fraction of new biomass
TAGDM	REAL	Total aboveground dry mass produced to date, neglecting losses due to leaf senescence (in g m^{-2})
TAVE	REAL	Daily average air temperature (in deg C)
TOLER	REAL	Total amount of error in the solution that can be tolerated without solution convergence
XAGDM	REAL	Initial biomass at crop emergence (in g m^{-2})
XD	REAL	Denominator in the parabolic interpolation equation
XDEF	REAL	Parameter default value input from the keyboard
XDELA	REAL	Initial bracketing search step size for parameter A
XDELB	REAL	Initial bracketing search step size for parameter B

XDELC	REAL	Initial bracketing search step size for parameter C
XGLAI	REAL	Initial GLAI at crop emergence
XN	REAL	Numerator in the parabolic interpolation equation
YES1	CHAR	Variable holding the uppercase symbol for "yes"
YES2	CHAR	Variable holding the lowercase symbol for "yes"
YF	REAL	Fraction of the daily biomass production partitioned to grain
YIELD	REAL	Final grain yield (in kg ha^{-1})
YINTC	REAL	XGLAI value calculated using the secant method for the next iteration in reinitialization

```
C     PROGRAM CONV_WX
      CHARACTER YES1,YES2,NO1,NO2,ANS,FMT(70)
      DIMENSION X(50)
      WRITE(*,*) '-----------------------------------------------'
      WRITE(*,*) '| This program converts the daily weather data  |'
      WRITE(*,*) '| in an existing data set into the format that  |'
      WRITE(*,*) '| is needed to run PCGRAMI. The output of this  |'
      WRITE(*,*) '| program will be a data file containing only   |'
      WRITE(*,*) '| the day of year, average daily temperature in |'
      WRITE(*,*) '| deg C, and the daily sum of solar radiation   |'
      WRITE(*,*) '| in MJ PAR per square meter per day.           |'
      WRITE(*,*) '-----------------------------------------------'
      WRITE(*,*) '  '
      DATA YES1,YES2,NO1,NO2/'Y','y','N','n'/
C
      KFREE = 1
      WRITE(*,*) 'EXISTING WEATHER DATA WILL BE READ FROM UNIT 10'
      OPEN(10,STATUS='OLD',FILE='   ')
      WRITE(*,*) 'CONVERTED WEATHER DATA WILL BE WRITTEN TO UNIT 20'
      OPEN(20,STATUS='NEW',FILE='   ')
C
      WRITE(*,*) 'ENTER THE YEAR IN WHICH THE WEATHER DATA START- '
      READ(*,*) IYEAR
      NYEAR = 365
      IF ((IYEAR/4)*4 .EQ. IYEAR) NYEAR = 366
      WRITE(*,*) 'ENTER THE DAY OF YEAR ON WHICH THE WEATHER DATA',
     2           ' START- '
      READ(*,*) ISTRT
C
      WRITE(*,*) 'ENTER THE NUMBER OF COLUMNS IN THE EXISTING DATA',
     2           ' SET (MAX OF 50)'
      WRITE(*,*) '(THIS IS EQUIVALENT TO THE NUMBER OF VARIABLES',
     2           ' IN THE DATA SET)- '
      READ(*,*) NCOL
C
      WRITE(*,*) 'INDICATE WHAT DAILY AIR TEMPERATURE VARIABLES',
     2           ' ARE IN THE EXISTING DATA SET:'
      WRITE(*,*) '  ENTER 1 FOR AVERAGE DAILY TEMPERATURE (TAVE),'
      WRITE(*,*) '  ENTER 2 FOR DAILY MAX AND MIN TEMPS (TMAX AND',
     2           ' TMIN),'
      WRITE(*,*) '  ENTER 3 FOR ALL THREE (TAVE, TMAX AND TMIN)- '
      READ(*,*) KTEMP
      IF (KTEMP .EQ. 2) THEN
         WRITE(*,*) 'ENTER THE COLUMN NUMBERS FOR TMAX AND TMIN'
         WRITE(*,*) '(SEPARATED BY A COMMA)- '
         READ(*,*) NCOLX,NCOLN
      ELSE
         WRITE(*,*) 'ENTER THE COLUMN NUMBER FOR TAVE- '
         READ(*,*) NCOLA
```

```
      ENDIF
      WRITE(*,*) 'ENTER THE COLUMN NUMBER FOR SOLAR RADIATION',
     2           ' DATA (RAD)-'
      READ(*,*) NCOLR
      WRITE(*,*) 'ARE ALL COLUMNS OF DATA IN THE EXISTING WEATHER',
     2           ' DATA SET SEPARATED'
      WRITE(*,*) 'BY BLANKS AND/OR COMMAS ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,1000) ANS
 1000 FORMAT(A1)
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) GO TO 2000
      KFREE = 0
      WRITE(*,*) 'ENTER THE INPUT FORMAT FOR THE COLUMNS OF DATA',
     2           ' IN THE EXISTING DATA SET,'
      WRITE(*,*) 'ENCLOSED IN PARENTHESES LIKE-'
      WRITE(*,*) '(F5.1,F7.2,F10.3,F5.0)'
      WRITE(*,*) '-ASSUME ALL VARIABLES ARE REALS, AND USE THE',
     2           ' FORMAT Fn.0 FOR INTEGER VALUES,'
      WRITE(*,*) ' WHERE n IS THE NUMBER OF DIGITS IN THE INTEGER',
     2           ' VARIABLE-'
      READ(*,1010) (FMT(I), I=1,70)
 1010 FORMAT(70A1)
C
 2000 WRITE(*,*) 'INDICATE THE UNITS OF TEMPERATURE IN THE',
     2           ' EXISTING DATA SET:'
      WRITE(*,*) '  ENTER 1 FOR DEG FAHRENHEIT,'
      WRITE(*,*) '  ENTER 2 FOR DEG CELSIUS,'
      WRITE(*,*) '  ENTER 3 FOR SOME OTHER UNITS-'
      READ(*,*) KUNITT
      IF (KUNITT .EQ. 3) THEN
         WRITE(*,*) 'UNITS OF TEMP MUST BE EITHER DEG F OR C-'
         WRITE(*,*) '  CONVERSION ATTEMPT ABORTED'
         GO TO 9999
      ENDIF
      WRITE(*,*) 'INDICATE THE UNITS OF SOLAR RADIATION IN THE',
     2           ' EXISTING DATA SET:'
      WRITE(*,*) '  ENTER 1 FOR MJ/Sq.M/DAY FULL SOLAR SPECTRUM,'
      WRITE(*,*) '  ENTER 2 FOR MJ/Sq.M/DAY PAR,'
      WRITE(*,*) '  ENTER 3 FOR LANGLEYS/DAY FULL SOLAR SPECTRUM,'
      WRITE(*,*) '  ENTER 4 FOR LANGLEYS/DAY PAR,'
      WRITE(*,*) '  ENTER 5 FOR SOME OTHER UNITS-'
      READ(*,*) KUNITR
      IF (KUNITR .EQ. 5) THEN
         WRITE(*,*) 'UNITS OF RAD MUST BE EITHER MJ/Sq.M OR',
     2              ' LANGLEYS PER DAY-'
         WRITE(*,*) '  CONVERSION ATTEMPT ABORTED'
         GO TO 9999
      ENDIF
C
      DO 5000 I=1, NYEAR
      IDAY = I + ISTRT -1
      IF (IDAY .GT. NYEAR) IDAY = IDAY -NYEAR
      IF (KFREE .EQ. 1) THEN
         READ(10,*,END=5001) (X(J), J=1,NCOL)
```

```
      ELSE
         READ(10,FMT,END=5001) (X(J), J=1,NCOL)
      ENDIF
      IF (KTEMP .EQ. 2) THEN
         TAVE = (X(NCOLX)+X(NCOLN))/2.
      ELSE
         TAVE = X(NCOLA)
      ENDIF
      RAD = X(NCOLR)
      IF (KUNITT .EQ. 1) TAVE = (TAVE-32.)/1.8
      IF (KUNITR.EQ.1 .OR. KUNITR.EQ.3) RAD = RAD*0.45
      IF (KUNITR.EQ.3 .OR. KUNITR.EQ.4) RAD = RAD*0.0418
      WRITE(20,2050) IDAY,TAVE,RAD
 2050 FORMAT(I5,2F10.2)
 5000 CONTINUE
C
 5001 WRITE(*,*) 'WEATHER DATA CONVERSION FINISHED'
 9999 STOP
      END
```

Appendix C: Listing of CONV_OB

```
C     PROGRAM CONV_OB
      CHARACTER YES1,YES2,NO1,NO2,ANS,FMT(70)
      DIMENSION X(50),OBS(100),IDT(100),IMO(12),IM2(12)
      WRITE(*,*) '---------------------------------------------'
      WRITE(*,*) '| This program extracts observations of leaf  |'
      WRITE(*,*) '| area index (LAI) from an existing data set  |'
      WRITE(*,*) '| and formats the observation date and LAI    |'
      WRITE(*,*) '| value in the form anticipated by PCGRAMI.   |'
      WRITE(*,*) '---------------------------------------------'
      WRITE(*,*) '   '
      DATA YES1,YES2,NO1,NO2/'Y','y','N','n'/
C
      KFREE = 1
      WRITE(*,*) 'EXISTING DATA FILE WILL BE READ FROM UNIT 10'
      OPEN(10,STATUS='OLD',FILE='   ')
      WRITE(*,*) 'CONVERTED DATA WILL BE WRITTEN TO UNIT 20'
      OPEN(20,STATUS='NEW',FILE='   ')
C
      WRITE(*,*) 'ENTER THE YEAR IN WHICH THE OBSERVATIONS START- '
      READ(*,*) IYEAR
      NYEAR = 365
      IF ((IYEAR/4)*4 .EQ. IYEAR) NYEAR = 366
      NYEAR2 = 365
      IF (((IYEAR+1)/4)*4 .EQ. (IYEAR+1)) NYEAR2 = 366
      WRITE(*,*) 'ENTER THE DAY OF YEAR ON WHICH OBSERVATIONS',
     2          ' START- '
      READ(*,*) ISTRT
C
      WRITE(*,*) 'ENTER THE NUMBER OF COLUMNS IN THE EXISTING DATA',
     2          ' SET (MAX OF 50)'
      WRITE(*,*) '(THIS IS EQUIVALENT TO THE NUMBER OF VARIABLES',
     2          ' IN THE DATA SET)- '
      READ(*,*) NCOL
C
      WRITE(*,*) 'INDICATE HOW THE OBSERVATION DATE IS RECORDED',
     2          ' IN THE EXISTING DATA SET:'
      WRITE(*,*) '  ENTER 1 FOR DAY OF YEAR (JULIAN DAY),'
      WRITE(*,*) '  ENTER 2 FOR MONTH AND DAY (EXPRESSED AS',
     2          ' NUMBERS),'
      WRITE(*,*) '  ENTER 3 FOR SOME OTHER WAY- '
      READ(*,*) KDATE
      IF (KDATE .EQ. 3) THEN
         WRITE(*,*) 'OBS DATE MUST BE EITHER DAY OF YEAR OR',
     2             ' MONTH AND DAY- '
         WRITE(*,*) '  CONVERSION ATTEMPT ABORTED'
         GO TO 9999
      ENDIF
      IF (KDATE .EQ. 1) THEN
         WRITE(*,*) 'ENTER THE COLUMN NUMBER FOR OBSERVATION DATE- '
```

```
         READ(*,*) NCOLJ
      ELSE
         WRITE(*,*) 'ENTER THE COLUMN NUMBERS FOR MONTH AND DAY',
     2              ' OF OBSERVATION'
         WRITE(*,*) '(SEPARATED BY A COMMA)-'
         READ(*,*) NCOLM,NCOLD
      ENDIF
      WRITE(*,*) 'ENTER THE COLUMN NUMBER FOR LEAF AREA INDEX-'
      READ(*,*) NCOLL
      WRITE(*,*) 'ARE ALL COLUMNS OF DATA IN THE EXISTING DATA',
     2           ' SET SEPARATED'
      WRITE(*,*) 'BY BLANKS AND/OR COMMAS ?'
      WRITE(*,*) '(ENTER Y OR N)-'
      READ(*,1000) ANS
 1000 FORMAT(A1)
      IF (ANS.EQ.YES1 .OR. ANS.EQ.YES2) GO TO 2000
      KFREE = 0
      WRITE(*,*) 'ENTER THE INPUT FORMAT FOR THE COLUMNS OF DATA',
     2           ' IN THE EXISTING DATA SET,'
      WRITE(*,*) 'ENCLOSED IN PARENTHESES LIKE-'
      WRITE(*,*) '(F5.1,F7.2,F10.3,F5.0)'
      WRITE(*,*) '-ASSUME ALL VARIABLES ARE REALS, AND USE THE',
     2           ' FORMAT Fn.0 FOR INTEGER VALUES,'
      WRITE(*,*) ' WHERE n IS THE NUMBER OF DIGITS IN THE INTEGER',
     2           ' VARIABLE-'
      READ(*,1010) (FMT(I), I=1,70)
 1010 FORMAT(70A1)
C
 2000 NOBS = 0
      DATA IMO/0,31,59,90,120,151,181,212,243,273,304,334/
      DATA IM2/0,31,60,91,121,152,182,213,244,274,305,335/
      DO 3000 I=1, 12
      K = 13 -I
      IF (NYEAR .EQ. 365) THEN
         NX = ISTRT -IMO(K)
      ELSE
         NX = ISTRT -IM2(K)
      ENDIF
      MONST = K
      IF (NX .GT. 0) GO TO 3001
 3000 CONTINUE
 3001 CONTINUE
C
      DO 5000 I=1, 100
      IF (KFREE .EQ. 1) THEN
         READ(10,*,END=5001) (X(J), J=1,NCOL)
      ELSE
         READ(10,FMT,END=5001) (X(J), J=1,NCOL)
      ENDIF
      IF (KDATE .EQ. 2) THEN
         MONTH = IFIX(X(NCOLM) +.5)
         IF (MONTH .GE. MONST) THEN
            IF (NYEAR .EQ. 365) THEN
               IDAY = IFIX(X(NCOLD)+.5) + IMO(MONTH)
```

```
            ELSE
               IDAY = IFIX(X(NCOLD)+.5) + IM2(MONTH)
            ENDIF
         ELSE
            IF (NYEAR2 .EQ. 365) THEN
               IDAY = IFIX(X(NCOLD)+.5) + IM0(MONTH) + NYEAR
            ELSE
               IDAY = IFIX(X(NCOLD)+.5) + IM2(MONTH) + NYEAR
            ENDIF
         ENDIF
      ELSE
         IDAY = IFIX(X(NCOLJ) +.5)
         IF (IDAY .LT. ISTRT) IDAY = IDAY + NYEAR
      ENDIF
      NOBS = NOBS + 1
      IDT(I) = IDAY
 5000 OBS(I) = X(NCOLL)
C
 5001 CONTINUE
      DO 7000 K=1, NOBS
         DO 6000 I=K, NOBS
         IF (IDT(I) .LT. IDT(K)) THEN
            ITEMP = IDT(K)
            IDT(K) = IDT(I)
            IDT(I) = ITEMP
            OTEMP = OBS(K)
            OBS(K) = OBS(I)
            OBS(I) = OTEMP
         ENDIF
 6000    CONTINUE
      IF (IDT(K) .GT. NYEAR) IDT(K) = IDT(K) - NYEAR
 7000 WRITE(20,7050) IDT(K),OBS(K)
 7050 FORMAT(I5,F10.3)
C
      WRITE(*,*) 'DATA CONVERSION FINISHED'
 9999 STOP
      END
```

Appendix D: Parameter Default File

PARAMETER DEFAULT VALUES FOR WINTER WHEAT

VALUE	PARAMETER	
100.	Sum Deg-Days to Emergence	(SDDEM)
1200.	Sum Deg-Days to Anthesis	(SDDAN)
1850.	Sum Deg-Days to Maturity	(SDDPM)
0.	Base Temp for Deg-Days	(BASET)
0.01	Initial Leaf Area Index	(XGLAI)
0.6	Light Extinction Coefficient	(EX)
2.5	DM/PAR Conversion Ratio	(EF)
0.024	Specific Leaf Area	(SLA)
0.9	Grain DM Partitioning Ratio	(YF)
0.088	Initial Value for Parameter A	(A)
0.003	Initial Value for Parameter B	(B)
70.	Init Value for Leaf Lifespan	(C)
0.001	Init Iter Interval for A	(XDELA)
0.0001	Init Iter Interval for B	(XDELB)
4.	Init Iter Interval for C	(XDELC)
0.0001	Converg Criterion for A	(CONVA)
0.00001	Converg Criterion for B	(CONVB)
0.5	Converg Criterion for C	(CONVC)
0.01	Conv Criterion for XGLAI	(CONVG)
0.05	Average Tolerable Error	(AVTOL)